OCOPOMO
Open Collaboration in Policy Modelling

D2.1 PLATFORM ARCHITECTURE AND FUNCTIONAL DESCRIPTION OF COMPONENTS

Document Full Name	**OCOPOMO_D2.1_v1.0**
Date	**20/12/2010**
Work Package	**WP2: Architectural design of IT solution**
Lead Partner	**TUK**
Author	Marian Mach, Melanie Bicking, Peter Butka, Karol Furdik, Jan Genci, Marta Kacprzyk, Ruth Meyer, Scott Moss, Magda Roszczynska, Sabrina Scherer, Stefan Ventzke, Maria A. Wimmer
Document status	**v1.0**
Dissemination level	**PUBLIC**

Document Log

Version	Date	Comment	Authors (alphabetical order)
001	04/08/2010	Initial version of SOTA on integration methods and technologies	Peter Butka
002	18/08/2010	SOTA on e-participation tools and technologies	Sabrina Scherer, Stefan Ventzke
003	26/08/2010	SOTA on formal modelling tools and technologies	Ruth Meyer
004	11/09/2010	Update of SOTA on formal modelling tools and technologies	Ruth Meyer, Scott Moss
005	13/09/2010	SOTA on scenario generation	Melanie Bicking
006	17/09/2010	Update of SOTA on e-participation tools and technologies	Sabrina Scherer
007	21/09/2010	Update of SOTA on scenario generation	Melanie Bicking
008	22/09/2010	Two new user requirements	Sabrina Scherer
009	23/09/2010	Architecture design methodology	Marian Mach
010	28/09/2010	Update of SOTA on integration methods and technologies	Peter Butka
011	29/09/2010	SOTA on relevant standards	Karol Furdik
012	05/10/2010	Correction of SOTA on scenario generation	Melanie Bicking
013	08/11/2010	Draft chapter about system boundaries, update of SOTA on scenario generation, and section on overview of tools for the OCOPOMO policy process	Melanie Bicking
014	08/11/2010	Usability perspective draft, updated CMS section	Sabrina Scherer, Maria A. Wimmer
015	10/11/2010	Section on User-oriented process perspective on OCOPOMO system	Magda Roszczynska
016	10/11/2010	Update of SOTA on formal modelling tools and technologies	Scott Moss
017	11/11/2010	Functional view – overall functional architecture	Peter Butka
018	12/11/2010	Rule manager functional description	Marian Mach
019	15/11/2010	Update of Functional view	Marian Mach
020	15/11/2010	Update of User-oriented process perspective on OCOPOMO system	Magda Roszczynska

021	15/11/2010	Update of Integration methods and technologies	Peter Butka
022	16/11/2010	Simulation manager functional description	Marian Mach
023	16/11/2010	Adding missing references, update of Usability perspective, Architecture validation methodology	Sabrina Scherer
024	16/11/2010	New user requirements based on use case analysis	Magda Roszczynska
025	18/11/2010	Update chapter about system boundaries	Melanie Bicking
026	19/11/2010	Annotation manager functional description	Marian Mach
027	19/11/2010	Interaction perspective and Mock-ups	Melanie Bicking
028	20/11/2010	Update of Overall functional architecture	Peter Butka
029	20/11/2010	Information view	Karol Furdik
030	22/11/2010	Update of SOTA on relevant standards	Karol Furdik
031	23/11/2010	Functional descriptions of Collaboration space, Search, Process, Notification, User, Concept and Link managers	Peter Butka
032	23/11/2010	Corrections in System boundaries, SOTA on scenario generation and analysis, User oriented process perspective	Melanie Bicking
033	23/11/2010	Update of Interaction perspective and Mock-ups	Melanie Bicking, Sabrina Scherer
034	23/11/2010	Comments on Usability perspective and Architecture validation	Sabrina Scherer
035	24/11/2010	Discussion forums manager functional description	Jan Genci
036	26/11/2010	Functional description of Chat, Calendar, Polling and Rating as well as Content managers	Peter Butka
037	26/11/2010	Functional description of Version and Document managers	Jan Genci
038	27/11/2010	Update of System boundaries	Karol Furdik
039	29/11/2010	Update of Interaction perspective and mock-up for registration	Sabrina Scherer
040	29/11/2010	Internationalisation perspective	Peter Butka
041	30/11/2010	Update of Usability perspective	Sabrina Scherer
042	02/12/2010	Application of perspectives to relevant views	Peter Butka

043	03/12/2010	Update of Architecture validation	Peter Butka
044	03/12/2010	Update of simulation use case	Magda Roszczynska
045	03/12/2010	Conclusion	Peter Butka
046	07/12/2010	Consolidation, corrections and preparation for internal review	Marian Mach
047	17/12/2010	Revisions based on internal review	Peter Butka, Karol Furdik, Marta Kacprzyk, Marian Mach, Magda Roszczynska
100	20/12/2010	Preparing final version	Marian Mach

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS AND ACRONYMS

ABM	Agent-Based Modelling
AI	Artificial Intelligence
AMQP	Advanced Message Queuing Protocol
ANSI	American National Standards Institute
API	Application Programming Interface
APS	Application Server
BPEL	Business Process Execution Language
BPM	Business Process Management
BPMI	Business Process Management Initiative
BPMN	Business Process Modelling Notation
BSD	Berkeley Software Distribution
CAQDAS	Computer-Assisted Qualitative Data Analysis Software
CCD	Consistent Conceptual Descriptions
CMIS	Content Management Interoperability Services
CMS	Content Management System
COM	Component Object Model
CORBA	Common Object Request Broker Architecture
CSS	Cascading Style Sheets
CSV	Comma-Separated Values
CWT	Collaborative Writing Tools
DCOM	Distributed Component Object Model
EAI	Enterprise Application Integration
ECMS	Enterprise Content Management System
ESB	Enterprise Service Bus
EMS	Electronic Meeting Systems
FTP	File Transfer Protocol
GDSS	Group Decision Support Systems
GNU	GNU’s Not Unix
GPL	General Public Licence
GSS	Group Support Systems
HTML	HyperText Markup Language
HTTP(S)	HyperText Transfer Protocol (Secure)
ICT	Information and Communication Technology
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IIOP	Internet Inter-ORB Protocol
IPC	Inter Portlet Communication
ISO	International Organisation for Standardisation
IT	Information Technology
J2EE	Java 2 Enterprise Edition
JBI	Java Business Integration
JCA	J2EE Connector Architecture

JCP	Java Community Process
JCR	Java Content Repository
JDBC	Java DataBase Connectivity
JDO	Java Data Object
JMS	Java Message Service
JMX	Java Management Extension
JSR	Java Specification Request
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
LHS	Left-Hand Side
MDA	Model Driven Architecture
MEP	Message Exchange Pattern
MIME	Multipurpose Internet Mail Extensions
MOM	Message Oriented Middleware
NTLM	NT LAN Manager
OASIS	Organization for the Advancement of Structured Information Standards
ODBC	Open DataBase Connectivity
OLE	Object Linking and Embedding
OMG	Object Management Group
ORB	Object Request Broker
OWL-S	Web Ontology Language for Services
OSS	Open Source Software
PDF	Portable Document Format
PHP	Hypertext Preprocessor
QDA	Qualitative Data Analysis
QoS	Quality of Service
RBAC	Role-Based Access Control
RDF	Resource Description Framework
REST	REpresentational State Transfer
RHS	Right-Hand Side
RMI	Remote Method Invocation
RPC	Remote Procedure Call
RSS	Rich Site Summary / RDF Site Summary / Really Simple Syndication
SAML	Security Assertion Markup Language
SDK	Software Development Kit
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SOTA	State Of The Art
SMTP	Simple Mail Transfer Protocol
SQL	Structured Query Language
STOMP	Streaming Text Oriented Message Protocol
TC	Technical committee
TPM	Transaction Processing Monitor
UDDI	Universal Description, Discovery and Integration

UML	Unified Modelling Language
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
WAI	Web Accessibility Initiative
WCF	Windows Communication Foundation
WCMS	Web CMS
WfMC	Workflow Management Coalition
WS	Web Service
WSDL	Web Service Description Language
WS-I	Web Services Interoperability
WSML	Web Service Modelling Language
WSRP	Web Service for Remote Portlet
WYSIWYG	What You See Is What You Get
XHTML	eXtensible HyperText Markup Language
XML	eXtensible Markup Language
XML-RPC	eXtensible Markup Language Remote Procedure Call
XMPP	eXtensible Messaging and Presence Protocol
XSLT	Extensible Stylesheet Language for Transformations

EXECUTIVE SUMMARY

The deliverable forms one of the pillars on which the implementation work to be accomplished in subsequent workpackages will be based. It represents a link between needs and expectations elicited in WP1 [Bicking et al., 2010] and the implementation of a system able to fulfil these needs and to meet these expectations in WP3. The content of the deliverable can be roughly divided into three parts: (i) Context and process understanding; (ii) State of the art analysis; and (iii) Architecture development.

Context and process understanding summarises and elaborates partners' ideas on processes behind the approach to policy modelling adopted within the OCOPOMO project as well as on which parts of these processes are expected to be supported by the prospective OCOPONO ICT toolbox and in which way. This part consists of two sub-parts:

- System boundaries
- User oriented process perspective

System boundaries outline the scope of the prospective system as a system enabling to deal with three types of scenarios (initial scenario, evidence-based user generated scenarios, model-based scenarios) and supporting activities related to production and analysis of scenarios. It is complemented by the definition of context of the system to be developed which introduces external entities the system is expected to communicate with as well as data flows between the system and these entities.

User oriented process perspective presents a set of use case diagrams illustrating expected activities of system users. The following cases have been elaborated: Registration, Initiation of the project, Working with the project, Collaboration space, Scenario generation, Scenario analysis, Qualitative data analysis (including several sub-cases), Quantitative data analysis, Network visualisation, Policy modelling, Simulation, and Evaluation of simulation results. A comparison of the use cases and currently defined user requirements resulted in the definition of a set of new requirements.

State of the art has focused on the areas which are most relevant for the project: e-participation, scenario generation, scenario analysis, formal modelling, and integration. For each of these areas several steps have been performed:

- Alternative identification
- Criteria selection
- Tool evaluation

Since the project is trying to reuse existing software tools and to shift its focus on development of missing tools only, alternative tools have been identified for each considered area. Subsequently, in order to select an appropriate tool, if possible, a set of criteria has been defined. Based on these criteria, the selected alternative tools have been evaluated and evaluation results have been discussed to support or to reject possible reuses. This approach enabled us to select a few basic tools to base the OCOPOMO ICT toolkit on their integration and enhancement to provide functionality which is missing. In addition, a set of collected requirements has been enriched by a few new requirements inspired by the used selection criteria.

The architecture part outlines ideas of the consortium developer partners how the OCOPOMO ICT toolbox is going to be built regarding its internal module structure. For the architecture design process we have used an approach based on the work [Rozanski and Woods, 2005]. This part focuses on the following basic areas:

- Architectural views and perspectives
- Component functional description
- Architecture validation

Architectural views and perspectives represent the overall architecture description. In order to present this architectural description, a “divide and conquer” principle has been employed – the description has been partitioned in order to approach it from different points of view simultaneously. The architectural views represent particular aspects of the architecture. Based on characteristics of the prospective system, two views have been employed: functional and information. Both views include not only the description itself but design considerations presenting decisions on which the design is based as well. The architectural perspectives address particular quality properties of the architecture. Three perspectives have been incorporated into the design: internationalisation, interaction and usability perspectives.

The developed three tier architecture has been broken down into architectural components called managers. Altogether, seventeen managers have been defined: Annotation Manager, Calendar Manager, Chat Manager, Collaboration Space Manager, Concept Manager, Content Manager, Discussion Forums Manager, Document Manager, Link Manager, Notification Manager, Polling and Rating Manager, Process Manager, Rule Manager, Search Manager, Simulation Manager, User Manager, and Version Manager. The deliverable provides functional description of these managers. The information on each of them incorporates relevant user requirements (requirements the manager responds to), context of the manager (relationships of the manager with the other managers), supported use cases and functionality description (use cases and functionality exposed by the manager) as well as a sketch of manager’s API (if the manager provides services for the other managers).

The last part tries to validate the presented architecture based on user requirements (both requirements collected in the previous project stage as well as requirements newly defined in this deliverable). The validation is twofold – requirement coverage is checked and an example is given how a user scenario can be supported by the collaboration of the designed managers.

1. INTRODUCTION

Modern approaches to policy modelling consider different disciplines and integrate both global problems and policy issues by using qualitative and quantitative methodologies, processes and tools in a framework that takes into consideration social as well as economic trends and conditions. Policy modelling serves to express possible strategies and to investigate their potential consequences. By doing so, existing policy analysis, modelling and simulation, as well as visualisation approaches are studied towards their capacity to contribute to policy formulation with particular focus on computer-assisted approaches.

Policy modelling is the process of abstraction that includes policy analysis that lays the foundation for conceptual modelling and formal modelling whereby formal modelling grounds again the simulation. At the end of the process stands the visualisation of the policy model or the simulation. Hence, visualisation refers to interface techniques and tools that help to visualise and present relevant information and issues. At each step throughout the policy modelling process different stakeholders can be involved and therefore make great and new challenges and opportunities on the visualisation. Besides, the whole process is influenced by the organisation and the strategy which is behind policy modelling, as well as the context and environment in which policy modelling takes place.

1.1. THE PURPOSE OF THE DELIVERABLE

The central challenge of the OCOPOMO project is to integrate formal policy modelling, scenario generation, open collaboration supporting stakeholders' engagement in social and economic policy with ICT solutions.

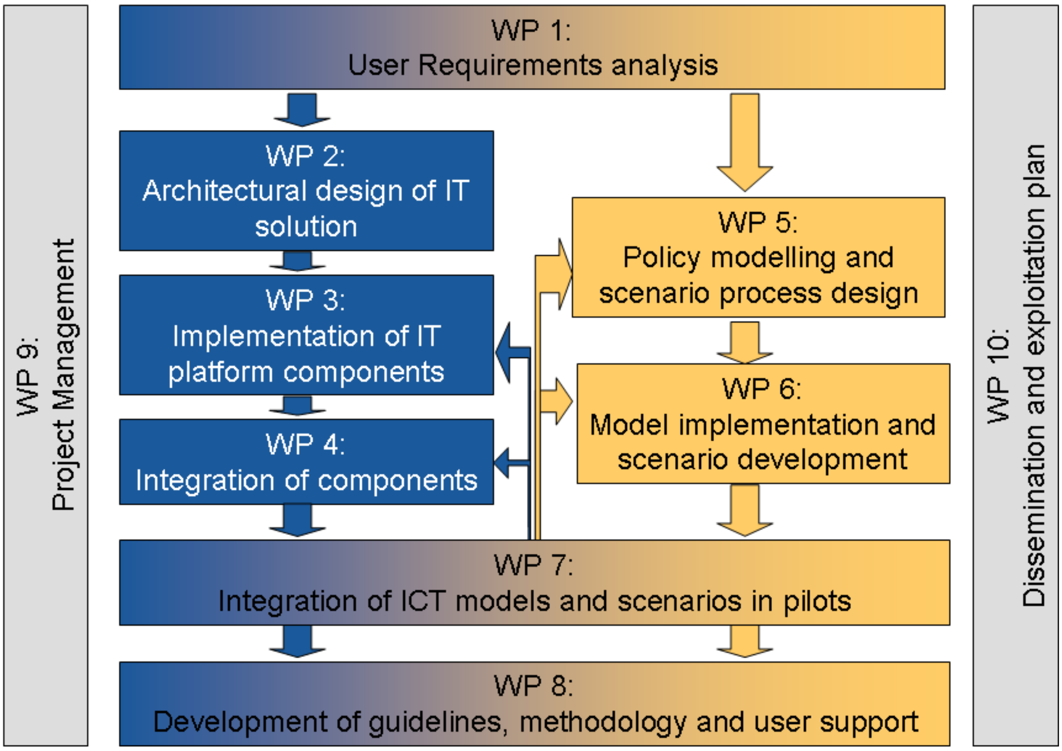

WP 1:
User Requirements analysis
WP 2:
Architectural design of IT solution
WP 3:
Implementation of IT platform components
WP 4:
Integration of components
WP 5:
Policy modelling and scenario process design
WP 6:
Model implementation and scenario development
WP 7:
Integration of ICT models and scenarios in pilots
WP 8:
Development of guidelines, methodology and user support
WP 9:
Project Management
WP 10:
Dissemination and exploitation plan

Figure 1 OCOPOMO's approach to implement the project

The project structure is depicted in Figure 1 where different colours indicate different project branches. One branch of the activities within the project (collected into workpackages with blue background) is to transform ideas on the process of policy modelling, its participants as well as information artefacts into an envisioned set of software tools – the OCOPOMO ICT toolbox. The toolbox is expected to support process participants in performing actions the process is composed from.

This deliverable represents a result of activities performed within workpackage WP2. Therefore, a description of policy modelling process along with a set of requirements provided by the previous workpackage WP1 [Bicking et al., 2010] represent a basis on which this deliverable (or its content) is built. A deep analysis of users' requirements was essential to guide subsequent work on extraction of relevant information and transforming this information into the presented content.

In order to make project activities consistent and enriching each other, all activities were performed in close collaboration with the ongoing workpackage WP5 [Moss et al. 2010].

In this deliverable, the focus is on the identification of various tools and technologies needed to support collaborative policy modelling as well as on their proper integration into a unified OCOPOMO system. Based on our initial understanding of processes behind policy modelling, two basic activity types were being performed:

- State of the art analysis – software categories relevant to policy modelling as intended within OCOPOMO have been identified and available tools and technologies have been investigated and evaluated (Task 2.1);
- Architecture development – proposing the whole system architecture (mainly from the functional and information views) as well as more detailed functional descriptions of all proposed architecture components (Tasks 2.2 and 2.3).

The former has resulted in the selection of a few software tools consistent with already collected requirements. The tools are expected to be reused in order to form a core of the prospective system. The latter has provided an architecture break-out into a set of basic software components. The main result presented in this deliverable is the definition of necessary system components and designation which of them should be developed from scratch and which should be prepared by reusing and/or modifying selected software tools as well as how these components should be integrated together into the OCOPOMO ICT toolkit. This information is expected to form input into subsequent workpackage WP3 focusing on implementation activities.

In addition to work presented within this deliverable, an environment for forthcoming implementation of the ICT toolbox has been identified and installed (Task 2.4). The developmental framework is documented on-line and can be found at the OCOPOMO web space[1]. It consists of a suite of tools for software design, coding and documenting, accompanied with the guidelines for commenting, coding and naming conventions. In accordance with the proposed technology approach, the implementation will be based on a specified version of Java IDE and Alfresco SDK. The code versioning, release control, and collaborative creation of system documentation will be supported by the shared code repository, bug tracking system, and central documentation environment.

[1] Path *Home – Workspace – WP02 – T2.4 Developmental framework*, direct link http://fgwimz3.uni-koblenz.de:8081/ocopomo/workspace/wp-02-architectural-design-of-it-solution-1/t2.4-developmental-framework

2. SYSTEM BOUNDARIES

The process of policy modelling, which is specifically addressed by OCOPOMO, is based on narrative scenarios and related formal policy models that are constructed and modified collaboratively, by various groups of involved persons that use proper e-participation tools for information exchange and mutual communication. To design the architecture of a software platform that will support this approach, the scope, context and boundaries of such system need to be specified as a basis for further development. In addition, groups of users – actors interacting with the system in particular phases of policy model creation should be identified together with their roles, competencies and responsibilities. Results of this initial analysis are presented in the following subsections and are used as a reference high-level functional description in the rest of this deliverable.

2.1. SCOPE OF THE SYSTEM

To outline the scope of the system to be developed, Figure 2 presents a set of information artefacts the prospective system has to manipulate with as well as a simplified control flow representing main actions to be performed by the system[2].

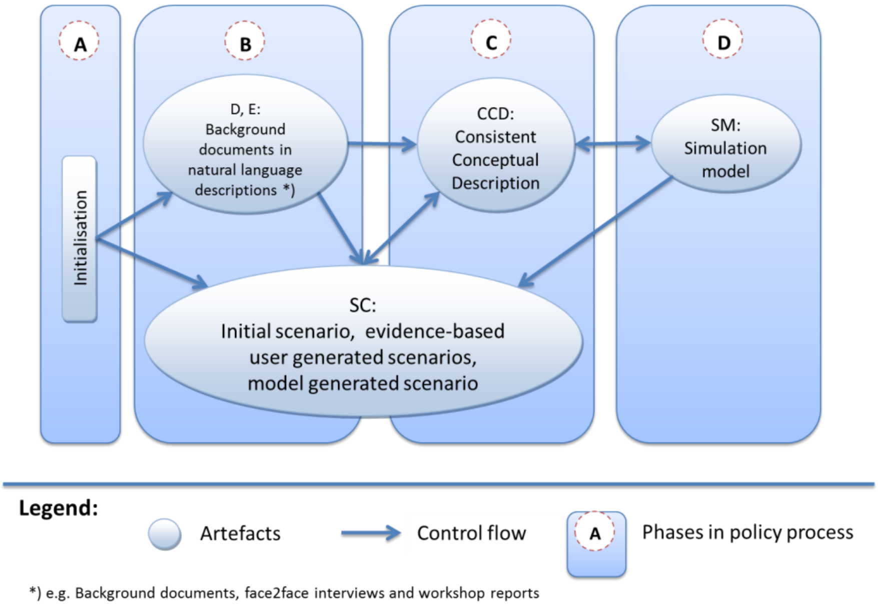

A
B
C
D
Initialisation
D, E:
Background documents in natural language descriptions *)
CCD:
Consistent Conceptual Description
SM:
Simulation model
SC:
Initial scenario, evidence-based user generated scenarios, model generated scenario
Legend:
Artefacts
Control flow
Phases in policy process
*) e.g. Background documents, face2face interviews and workshop reports

Figure 2 OCOPOMO main phases and information artefacts

The main artefact the whole project is based on is *Scenario*. Basically, it is in general a textual description (narrative, unstructured or structured text) of a perceived view or understanding of a topic under discussion. A scenario may cover an existing world status, mental models of stakeholders or an output of future simulations. Some features a scenario can posses:

- It may also depict a future vision, even some fiction.

[2] The overall methodological process described in more details is expected to be presented in an upcoming project deliverable D5.1 [Moss et al., 2010].

- Alternative scenarios may exist / be developed to describe different aspects and /or alternatives
- Different authors[3] may develop different sets of scenarios independently (reflecting e.g. different mental models in scenario sets of different groups).
- Some of the scenarios may also be conflicting among different author groups.
- Scenarios may be related each to other, for example a scenario can extend and therewith advance an existing scenario.

As indicated in the figure, it is possible to distinguish three types of scenarios, differing by their authors and place in the whole process:

- Initial scenario – provided usually by one person or a small group of people in order to stimulate the process of policy modelling and to set up a point of departure
- Evidence-based user generated scenario – collaboratively developed scenarios by human authors, communicating their opinions, views and expectations
- Model generated scenario – computed as a result of running a simulation model, produced as a text-based transcription of a simulation run

In order to produce scenarios, other information artefacts are dealt with as well. Two of them are simulation models and consistent conceptual descriptions. In addition to them, the process is supported by other background artefacts as documents and/or human experience.

A *Simulation Model* is a simplified abstract view of the complex reality thereby representing objects, phenomena, and processes in a logical way. When creating a simulation model, three elements are identified: the parts of the system, the interaction between the parts, and the number and nature of inputs. A model is essentially created for each of these, with crucial aspects considered and minor aspects ignored. Models can perform two fundamentally different representational functions: a) a model can represent a selected part of the world (the 'target system') or b) a model can represent a theory i.e. it interprets rules and axioms of that theory.

The *Consistent Conceptual Description (CCD)* serves to capture descriptions and perceptions of the stakeholders in a structured way and code this information, cluster it, condense it and further elaborate it to reflect a comprehensive consistent conceptual description of a policy case. The content can be e.g. stored in a database, which allows different extractions and visualisations of content (e.g. social network, rule-dependency graph, actor hierarchies, relationships, conditions, etc.) also as understandable visualizations for end users.

CCD plays a role of an intermediary between scenarios and simulation models. Several scenarios can form input to the CCD of a policy domain and further lead to a formal simulation model. The similar role is played by the CCD in analysing simulation models in order to update scenarios. The CCD filters and structures the information, and also guides the elicitation of further information that one may find useful to increase understanding of the domain in question. The envisioned system to be developed must allow going back and forth in the process of developing policy models (from scenarios to simulation models as well as from simulation models to scenarios – in both cases via CCD); hence the arrows are depicted in both directions. Links between scenarios and CCD as well as between CCD and simulation models need to be maintained with the aim of ensuring traceability back and forth.

[3] Differentiation of 'authors' and other involved user types is given in the next section.

The Figure 2 shows the control flow among the artefacts as well. The process flow among the artefacts was grouped into four phases (A – D):

- *Phase A* includes the initiation, where a user prepares a policy case to be discussed and developed. The user provides initial description of the policy case. This phase results in an initial scenario.
- *Phase B* enables interaction between users. Users provide background data/experience (E) and documents (D). Likewise, evidence-based user-generated scenarios are developed by the users.
- In *phase C*, CCD (e.g. topic maps, ontology, qualitative data analysis with knowledge structures such as social networks, rule-dependency graphs, etc.) are elaborated based on the evidence-based user generated scenarios and the inputs of background experience and documents.
- In *phase D*, the knowledge accumulated within CCD is transformed into simulation models. Model-based scenarios are generated as output from simulation models and are used as visualisation of the output to be communicated to users.

The user is to be supported by the prospective ICT toolbox in the different steps of the process, i.e. the system shall facilitate both understanding and analysis. The system shall support the integration of different user types at different points in time (i.e. stages) in the OCOPOMO process.

2.2. CONTEXT OF THE SYSTEM

The context diagram of the OCOPOMO system is introduced to present only the central OCOPOMO process that subsumes everything inside the scope of the OCOPOMO system. The context diagram depicted in Figure 3 shows how the system will receive and send data flows to the external entities (external entities represent prospective users of the OCOPOMO system) involved in the process of collaborative policy development.

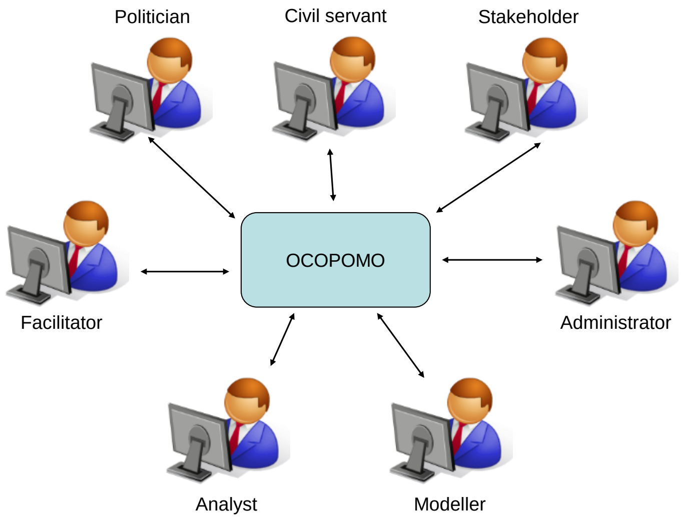


Figure 3 OCOPOMO system context diagram

The external entities correspond to the prospective users of the OCOPOMO system. Some of the proposed user types were already mentioned before, such as facilitator, stakeholder, etc. A more exact specification of the envisioned user roles, which correspond to the external entities from the context diagram, is provided in the following outline.

In general, two main groups of users can be distinguished:

1. *Direct participants of the policy development process*, who are intentionally involved in policy creation and have their own preferences, ideas, or proposals of how the newly created policy should look like. The group of direct process participants includes user roles such as:

   - *Politician*, a decision-maker and/or a person that is responsible for the policy implementation. Politicians may initiate collaborative policy development in OCOPOMO (directly, or through civil servants) and may participate on the development of narrative scenarios or policy models. It is supposed that politicians typically participate on the collaborative policy development in later phases when some results are already available.
   - *Civil servant*, an assistant of politicians and/or a provider of relevant supporting materials for other participants of the policy development process. Civil servants, together with politicians, may provide an initial scenario description, which serves as a starting point for collaborative development of a new or improved policy.
   - *Stakeholder*, end users such as citizens, NGO’s and SME’s, which are willing and able to participate actively in the construction of narrative scenarios, discussions, information exchange and other phases of the collaborative policy development.

2. *Actors that provide a methodological or technical support* for the policy development participants in the OCOPOMO collaborative environment. This group includes the following user roles:

   - *Facilitator*, a mediator, which methodologically controls the collaboration working space. Facilitators maintain the collaborative scenario development by providing initial text descriptions and uploading background documents referring the policy case. They are also responsible for inviting stakeholders of relevant interest groups, assigning user accounts, contacting analysts and modellers to provide respective models, controlling iteration cycles of narrative scenarios and publishing agreed policy descriptions.
   - *Analyst*, an expert that investigates scenarios and other (mostly textual) resources, analyses it and provides a formal representation of extracted knowledge. Analysts are responsible for the qualitative analyses of narrative scenarios, which result in the construction of CCD. The analysis includes an extraction of knowledge from discussions, comments, simulation results, and various materials that may support the development of scenarios.
   - *Modeller*, an expert that constructs formal policy models according to a given CCD. In other words, modellers derive the simulation models from an existing CCD, create the simulation environment and provide the constructed models for participants, which can run customisable simulations. Modellers are also responsible for maintenance of simulation results and their provision to analysts for enhancing the respective scenarios accordingly.
   - *Administrator*, responsible for technical maintenance of the system.

The user roles proposed for OCOPOMO differ from each other and, therefore, have different needs of support in the policy process and through the ICT such as, for example, different knowledge of the existing policy, principles of policy formation, and technical background.

The presented external entities (users of the system) communicate with the system in order to provide/obtain data. In order to describe data flows between external entities and the system more precisely, the description is broken down to respect three main activity areas the OCOPOMO approach focuses on – scenario generation, scenario analysis and transformation, and simulation, evaluation and validation.

Scenario Generation

The scenario generation is one of the main tasks. In OCOPOMO, the starting point of a policy case is an existing policy. In general, such a policy can be brought in either by a government agency (i.e. politician, civil servant) or by an interest group. Based on this policy one initial scenario description is generated. Then, stakeholders can generate further scenarios (scenario alternatives or scenarios of different groups reflecting e.g. conflicting views).

External entity	Direction	Description
Politician	In	existing policy, background documents, initial scenario description
	Out	initial scenario description, further scenarios (evidence-based user scenarios)
Civil servant	In	existing policy, initial scenario description
	Out	background documents
Stakeholder	In	initial scenario description, evidence-based user scenarios, background documents
	Out	further scenarios (evidence-based user scenarios), background documents

Table 1 Description of communication flows between external entities and the system during scenario generation

Scenario Analysis and Transformation

In order to close the gap between scenarios and simulation models, the process of transforming scenarios into simulation models may require the following structured information for creating the simulation model (parts of CCD):

- social networks (i.e. actors and dependencies),
- social processes,
- skill tables,
- conditions (evidences) and consequences (actions),
- if-then rules

Different kinds of scenarios are generated. First, the initial scenario is generated from the natural language descriptions. The initial scenario lays the foundation for the evidence-based user scenarios and gives first input for CCD. The evidence-based user scenarios enrich the CCD with further input. From the CCD the modellers derive simulation models, on which the simulation runs.

The structured information is derived from the master scenario document and the supportive data/documents, which are unstructured natural language descriptions. In this context, the social network is in particular important for the development of the simulation model as it presents structured relevant information on the agents and their relationships.

External entity	Direction	Description
Politician	In	social network, supportive data/documents
	Out	the master document, initial scenario, unstructured scenario alternatives, supportive data/documents
Civil servant	In	a request for providing supportive data or documents for scenario alternatives
	Out	supportive data/documents
Stakeholder	In	social network, unstructured scenario alternatives, the master document and supportive data/documents
	Out	enhanced unstructured scenario alternatives (final evidence-based user scenarios), the master document and the supportive data/documents (i.e. unstructured natural language descriptions)
Facilitator	In	initial scenario, unstructured scenario alternatives, the master document and the supportive data/documents
	Out	social network, requests for creating or enhancing CCD, rule dependency graphs and/or simulation models
Analyst	In	initial scenario, unstructured scenario alternatives (final evidence-based user scenarios), the master document and the supportive data/documents (i.e. unstructured natural language descriptions)
	Out	CCD, the rule dependency graph
Modeller	In	the master document and the supportive data/documents (i.e. unstructured natural language descriptions), CCD, the rule dependency graph
	Out	rules and the rule dependency graph, simulation model

Table 2 Description of communication flows between external entities and the system during scenario analysis and transformation

Simulation and Evaluation and Validation

Computers enable to run simulations based on the simulation model that covers relationships between the individual actions on the micro-level and the collective effects on the macro level to help understand interrelation and interdependencies and thereby making the system manageable.

The results received from simulations are visualised in a text format (i.e. a model-based scenario). Visualisation is needed to demonstrate how a strongly connected operation works and which results are generated and derivable on one side and to enable interaction in general on the other side. Visualisation is very important to provide results of a simulation to users and analysts as well as to receive feedback and interaction with those stakeholders. Describing a specified context’s concrete visualisation in detail will mostly include a direct link or at least a mention of the contextual simulation or the information source the present visualisation provides and works on.

The model-based scenario is compared with the evidence-based user scenarios for evaluation and validation.

External entity	Direction	Description
Analyst	In	simulation model, simulation results, model-based scenario, evidence-based user scenarios, CCD, rules and the rule dependency graph
	Out	simulation results re-visualised in a text format (i.e. one model-based scenario), enhanced CCD and/or the rule dependency graph, evaluation and validation
Facilitator	In	simulation model, model-based scenario, evidence-based user scenarios, simulation results re-visualised in a text format (i.e. one model-based scenario)
	Out	requests for creating or enhancing scenarios, CCD, rule dependency graphs and/or simulation models
Modeller	In	model-based scenario, CCD, evidence-based user scenarios
	Out	rules and the rule dependency graph, simulation model, simulation results, evaluation and validation

Table 3 Description of communication flows between external entities and the system during simulation and evaluation and validation

3. STATE OF THE ART ANALYSIS AND TECHNOLOGY IDENTIFICATION

Current project’s understanding of the policy modelling process enables to recognise essential building blocks this process consists of. Based on this, it was possible to identify main types of software tools/applications able to support users within particular modelling process steps. However, selection of the most suitable suite of software tools needs to be based on a detailed analysis of existing tools and technologies in the areas that should be covered by the OCOPOMO ICT toolbox. It namely includes various e-Participation tools, groupware frameworks, integration platforms, content management systems, tools for scenario building and analysis, technologies for formal modelling, multi-agent simulation and visualisation of rule-based policy models. The existing and available tool representatives of these areas are investigated, described and evaluated in the following subsections. Since the tools are expected to be integrated into one consistent toolbox, the focus is primarily on open source tools licences of which enable to reuse these tools for the project’s objectives.

In addition, a survey of standards (those standards that can be relevant to the approach of policy modelling adopted in OCOPOMO and thus should be considered during system design) is presented in Appendix A.

3.1. INTEGRATION METHODS AND TECHNOLOGIES

One important task for development of the OCOPOMO platform is to find a good solution for integration of all components. Different tools have to be incorporated into platform (in some way), mostly coming from basic parts of the project's elements like ICT tools for support of scenario generation process, e-participation tools, tools to support policy modelling, simulation tools. Analysis of integration methods and technologies for such applications is identified in this chapter, together with a connection to integration-related user requirements acquired during work on the project’s D1.1 deliverable (output of WP1) [Bicking et al., 2010].

3.1.1. Integration of software applications

In practice, many systems are not developed from the scratch but (at least partially) are integrated from existing applications. Research and technological fields related to Enterprise Application Integration (EAI) or (message-based) middleware integration solutions (also known as Message-Oriented Middleware - MOM) are those which fulfil all aspects of software and computer systems architectural principles in order to integrate a set of (enterprise) computer applications. Most of the well-known approaches follow the paradigm of Service-Oriented Architecture (SOA). Usually, if system development needs to be done by integration of several components, integration methods become more crucial, also if we want to produce new software by combining with other, as it is in our case of the OCOPOMO platform.

Integration is a difficult task in process of system development. In general, an integration project should answer three basic issues [Juric et al., 2007]: 1) Definition of integration architecture; 2) Selection of integration infrastructure and technologies; 3) Development and maintenance of integration documentation. In this part of the deliverable we want to describe mainly the possible solutions, general integration approaches, methods and overview of technologies in order to have a solution for the integration of the OCOPOMO platform.

3.1.1.1. Basic types/approaches to integration in general

There are basically two approaches to integrate several components (applications) - bottom-up and top-down. In the former case, a problem of communication of components is processed directly between them and problems are fulfilled from the scratch (where necessary). In case of the top-down approach the solution is based on logical, high-level integration architecture (without seeing components in details), where integration methods and processes are solved first. The top-down approach is preferred in those cases, where we have many components, quite different technologies and components could change during the development.

Sound integration architecture usually provides several benefits, like reusability, encapsulation, possible distribution of services, partitioning (build on specific tiers - middle, back-end, front-end), scalability, enhanced performance, improved reliability, manageability, increased consistency and flexibility, multiple clients support, independent and rapid development, better composition and configuration, improved security, etc.

Integration architecture is usually built in several systematic layers. Omitting a layer in such architecture is a short-term solution, but sometimes can emerge into new problems later. The most important types of integration are [Juric et al., 2007]:

- *Data-level integration* - focuses on moving data between applications with the objective of sharing the same data among different applications. Data-level integration is a relatively simple approach and often used as a starting solution (e.g. easily understood by developers, accessing databases is easy, several tools available for data sharing), does not require changes to the applications. The difficulties of data integration are in complexity of the databases and in their quantity. It is necessary to understand the data stored in databases and their structure. Semantics of the data stored in databases is the most difficult part of the data-level integration.
- *Application integration* - aims at sharing functionality (business logic), not just pure data, usually achieved through the use of application programming interfaces (APIs). The objective of application integration is to understand and use APIs for accessing the required functionalities and to mask the technology differences between different technologies used for APIs and their access (the latter is achieved using services). Interfaces provide one-way contracts between the applications. As long as the interfaces stay unchanged, this means that the contracts have not been changed. Good interfaces are loosely coupled - achieved by sharing integration-specific data (without behaviour), structuring the data and using open standard technologies for APIs.
- *Business process integration* - enables support for business processes where existing solutions take part in distinctive steps, exposes the functionality as abstractions of business methods through interfaces, existing applications are remodelled to expose the functionality of the business process tier and different pieces are glued together, usually by using a business process modelling and execution language. Advantage of such approaches is flexibility and adaptability to business process changes. Disadvantage is in business process reengineering and implementation of several specific technical layers for working at a higher-level of process abstraction.
- *Presentation integration* – existing applications are encapsulated and offered through high-level interfaces. Next logical step is that user gets unified view of the information system in one presentation layer hiding background applications and different executing of functions. The presentation integration is a step in which a common user interface (usually a portal) is

defined for the business-method-level integrated information system. It is a last piece of multi-tier integration architecture.

3.1.1.2. Integration infrastructure

The required infrastructure services for integration should be identified and separated into two different types - basic infrastructure services (useful for the majority of applications, if needed) and task-specific services (provide functionalities related to a specific task within infrastructure). In the first case we have four basic layers of services:

- *Communication* - provides the abstraction for communication details and transparency for accessing different remote systems and unifies the view on them. It enables the separation of business logic and the communication services, but allowing communication between them. Different types of middleware provide different services for communication within this layer like Database access technologies (for accessing and unifying of database connections), Message-Oriented Middleware (MOM, asynchronous communication through sending and receiving messages through a message queue or a message channel), Remote Procedure Call (RPC, communication services for synchronous, procedural-oriented communication, similar to object request brokers), or Enterprise Services Bus (ESB, integration broker targeted to fulfil the objectives of SOA).
- *Brokering / Routing* - most important for implementing the technical side of integration, adapts the communication between applications in order to fulfil interoperability of all applications. Responsibilities of this layer are in gathering required data from multiple sources (aggregation), preparing the data for processing in different applications (transformation), gathering results, and combining results (synthesis) with consistent presentation of them. To achieve this, the layer needs metadata information about particular applications, methods, messages, and interfaces, and the sequence of operations involved.
- *Transformation* – provides an engine (usually based on XSLT - Extensible Stylesheet Language for Transformations[4]) for easy specification of data and schema transformations, specifying transformation rules, templates. Advantage is that XSLT can be executed independently to programming language, platform, and other restrictions. These tools are becoming part of development environments and integration technologies (like ESB).
- *Business intelligence* - responsible for presenting a high-level interface to access business information to other applications and to the users using presentation tier, today mostly personalized portals.

In the second case (task-specific services) we can have several additional layers like:

- *Transactions* - business operations are carrying out in a transactional manner, any operation guarantees that the consistency of the system is preserved. It also has to isolate operations from other operations to a certain degree and guarantee that the outcomes of operations are written to the persistent storage.
- *Security* - provides ways to constrain access to the system. Security should include all four basic layers (also called horizontal), should be able to reuse the existing application security,

[4] XSLT specification - http://www.w3.org/TR/xslt

use roles, single user login, and it is related to aspects like communication channel encryption, authentication, authorization, and auditing.

- *Lifecycle* - provides ways to control the lifecycle of all applications, with easy replacement. It is important to minimize the dependencies between applications and specify ways for the applications to interoperate.
- *Naming* - unified naming service, usually implemented with a naming and directory product that enables storing and looking for name-related information.
- *Scalability* - integration infrastructure should be designed with scalability in mind, with concurrent access to applications, load balancing, performance and load tests, etc.
- *Management* - provides ways to manage the integration infrastructure, methods and tools to manage horizontal and vertical services, with easy configuration (declarative) and version management, best with a possibility for remote management access.
- *Rules* - definition of declarative rules for performing communication, brokering, routing, and business-intelligence tasks, like data formats, data transformations and flows, events, information processing, and information representation.

In practice integration problems and solutions are often identical or similar. For that reason well-known and reusable solutions can be classified into common types - integration patterns. Each integration pattern defines a common integration problem and a sound solution. The most important integration patterns are Integration broker (integration messenger), Wrapper (integration adapter, integration connector), Integration mediator, Single-step application integration, Multi-step application integration, Virtual service (integration facade), Data access object (data exchange pattern), Data mapping (standard, direct, multi-step), Process automation, etc. Many integration patterns can be found in various catalogues and books, e.g. [Hohpe and Woolf, 2003] or [Juric et al., 2002].

3.1.2. Overview of integration technologies

Integration infrastructure usually requires more than one technology (mixture of technologies). In this case interoperability of them is important. Integration can be difficult even for technologies based on open standards. Technologies used for integration are often called middleware - system services software that works between the operating system layer and the application layer and provides services. Middleware connects applications and provides connectivity and interoperability to the applications, and all forms are helpful in easing the communication between different software applications.

The selection of middleware affects architecture due to centralisation of software infrastructure and introduction of abstraction layer, which reduces the complexity. Disadvantage could be in communication overhead within the system, which can influence some efficiency factors (performance, scalability, etc.). This should be also considered in selection and architecture design process.

A lot of technologies are available as middleware products (solutions) for different integration approaches and methods. Now we will provide short overview of them.

3.1.2.1. Database access technologies

The simplest way for data-level integration is based on database access technologies, which is important for accessing and unifying of database connections. It means that we have an abstraction layer which provides access to the database(s) and enables us to change the actual data without modifying the application source code. Database access technologies are useful for extracting data from different databases. Basically, technologies differ in the form of interfaces to databases:

- Function-oriented access – used for accessing functions of database by some driver based on the unified language. Databases (like open-source solutions MySQL[5], PostgreSQL[6], etc.) have driver connectors with API for querying and updating their tables using SQL queries, where some standard is used. A well-known standard is ODBC (Open DataBase Connectivity)[7], in case of Java platform JDBC (Java DataBase Connectivity, latest in version 4.0)[8] is used.
- Object-oriented access – used for accessing objects from a database. Communication is done in more objective way using object-relational mapping. A basic feature is transparency of the persistent services to the domain model. JDO (Java Data Objects, latest in 2.2)[9] is a specification of Java objects persistence. In Microsoft .NET platform ADO.NET (ActiveX Data Objects for .NET)[10] fulfils this option. Object-relational mapping could be also accomplished using some specialised library, which extends function-oriented access with XML configuration or code annotations (e.g. Java annotations). This can be a very effective solution for mapping from objects to tables. One well-known solution in Java platform is Hibernate[11], which in latest version also supports .NET platform.

3.1.2.2. Message-oriented middleware

Message-oriented middleware (MOM) is a client/server infrastructure that enables and increases interoperability, flexibility, and portability of applications. It enables communication between applications over distributed and heterogeneous platforms and reduces complexity due to hiding of many details. APIs are used for functionality access. One of the basic characteristics is its asynchronous communication and use of message queues, where messages are able to contain any type of data and communication continues even if the receiver is temporary not available (wait for availability). Disadvantage of asynchronous communication is overloading.

MOM products are usually proprietary products and must specifically run on each and every platform being integrated. Java platform provides ways to achieve relatively high independence from a specific vendor through Java Messaging Service (JMS)[12], which is implemented by most vendors. AMQP – Advanced Message Queuing Protocol [Vinoski, 2006] is an emerging standard that defines the protocol and formats used in the messaging server and client, Java applications with AMQP are

[5] MySQL – http://mysql.com/

[6] PostgreSQL – http://www.postgresql.org/

[7] ODBC – http://www.openlinksw.com/info/docs/odbcwhp/tableof.htm

[8] JDBC – http://jcp.org/aboutJava/communityprocess/final/jsr221/index.html

[9] JDO – http://jcp.org/aboutJava/communityprocess/mrel/jsr243/index2.html

[10] ADO.NET on MSDN – http://msdn.microsoft.com/en-us/library/aa286484.aspx

[11] Hibernate – http://www.hibernate.org/

[12] JMS – http://www.oracle.com/technetwork/java/index-jsp-142945.html

typically written in JMS. There are also other standards available or under development like XMPP (Extensible Messaging and Presence Protocol)[13], STOMP (Streaming Text Oriented Message Protocol)[14], or RestMS[15] (similar to AMQP, but based on the RESTful HTTP).

3.1.2.3. Remote procedure calls

Remote procedure call (RPC) is also a client/server infrastructure similar to MOM, but with synchronous communication (request-reply), which blocks the client until the server fulfils the request. To achieve remote communication, applications use procedure calls. RPC guards against overloading a network. RPC increases the flexibility of architecture by allowing a client of an application to employ a function call to access a server on a remote system. RPC is appropriate for client/server applications in which the client can issue a request and wait for the server to return a response before continuing with its own processing, but requires that the recipient is on-line to accept the remote call.

Main idea under RPC is related to Distributed Computing Environment (developed by Open Systems Foundation). It is a set of integrating services for expanding RPC functionality: it provides directory, time, security, data-sharing and thread services. Many implementations of RPC protocols come from ONC/RPC specification[16]. Also many analogical systems for RPC exist, like Java RMI (Remote Method Invocation)[17], RPyC (RPC for Python)[18], .NET Remoting[19], etc. Interesting solution for RPC (lately extended into more complex SOAP and web services solutions) is protocol that uses XML to encode its calls and HTTP as a transport mechanism - XML-RPC[20].

3.1.2.4. Object request brokers

Object request broker (ORB) is another technology for achieving interoperability of applications that manages and supports the communication between distributed objects or components. ORBs provide transparency (independence) on location, programming language, protocol and operating system. Interfaces are used for communication between objects, where communication is synchronous (usually) or asynchronous. Location services are used for locating the components within network. In practice, ORBs provide all the components as local. This is good for development, but can influence the performance. ORB products have more options for implementing the functionality - moving (some) functions to the client and server components, provide them as a separate process, and integrate them into operating system. Main standards/solutions of ORBs are:

- CORBA[21] - Common Object Request Broker Architecture and compliant standards, which are using IIOP (Internet Inter-ORB Protocol)[22] for communication between components. CORBA

[13] XMPP – http://tools.ietf.org/html/rfc3920

[14] STOMP – http://stomp.codehaus.org/

[15] RestMS – http://www.restms.org/

[16] RPC: Remote Procedure Call Protocol Specification Version 2 – http://tools.ietf.org/html/rfc5531

[17] Java RMI – http://www.oracle.com/technetwork/java/javase/tech/index-jsp-136424.html

[18] RPyC – http://rpyc.wikidot.com/

[19] .NET Remoting on MSDN – http://msdn.microsoft.com/en-us/library/kwdt6w2k%28VS.71%29.aspx

[20] XML-RPC – http://www.xmlrpc.com/spec

[21] CORBA – http://www.omg.org/spec/CORBA/3.1/

[22] IIOP – http://www.omg.org/technology/documents/formal/corba_iiop.htm

standard, originally created by OMG (Object Management Group), is well-known and quite generic standard for ORB solutions.

- Java RMI (Remote Method Invocation) and RMI-IIOP - provide architecture and implementations for Java platform. In general, Java context applications are available using API or Java-specific remote transfer protocol. Java non-context applications are available using CORBA implementations based on the RMI-IIOP (RMI over IIOP, RMI interfaces supporting most of the CORBA functionality).
- Microsoft OLE/COM/DCOM/COM+/.NET Remoting/WCF [McLean et al., 2002)] - several architectures, elements and standards provided within Microsoft platform for distributed applications and their communication. A series of technologies exists for supporting such functionality: Object Linking and Embedding (OLE), through Component Object Model (COM), Distributed COM version (known as ActiveX), COM+, Windows Communication Foundation (WCF, part of .NET 3.0).

3.1.2.5. Web services

Service oriented architectures (SOAs) are currently the most interesting topic in modern information systems development. Software systems adhering to the SOA paradigm provide the several main functionalities achieved by the web services [Papazoglou, 2003]: service *publication* (service descriptions are created in a suitable format and are published according to pre-defined standards in well-known locations), service *discovery* (uses information retrieval techniques on the published service descriptions), service *selection* (filters the results of the discovery process), and service *binding* (prepares the main execution of a service).

Web services, definable in general as "any service that is available over the Internet, uses a standardized XML messaging system, and is not tied to any operating system or programming language" [Cerami, 2002], are the latest distributed technology and provide the technological foundation for achieving interoperability between mentioned elements. The components for web services are standardized. In general three basic aspects are important:

- *XML messaging system* – most widely used implementations of XML messaging are SOAP (Simple Object Access Protocol)[23], XML-RPC (also mentioned in RPC section) and REST – REpresentational State Transfer [Fielding, 2000]. SOAP is a lightweight protocol intended for exchanging structured information in a decentralized, distributed environment [Gudgin et al., 2007]. SOAP basically works by tunnelling XML-formatted messages via Internet protocols (SMTP, HTTP(S)) and is easy for implementation in existing infrastructures. XML-RPC simplifies SOAP approach by restriction to HTTP(S), where content is transferred in a POST message. REST further simplifies the process by usage of intuitive request format directly based on the HTTP methods of GET, POST, PUT and DELETE. REST becomes very popular solution with SOA and many technologies start to support this standard
- *Self-description of services* – important for description of services in terms of available functions with expected input. Different standards have been created during time which can be grouped into two categories:

[23] SOAP – http://www.w3.org/TR/soap/

    - *Fundamental web service descriptions* – based on WSDL (currently in revision 2.0)[24]. It is a XML format which divides Web Services on two levels - abstract and concrete. The abstract one describes message types for exchanging, inputs and outputs, sequence of messages sent between client and server (MEP – message exchange pattern), all together can be viewed as interface. The concrete part adds the information needed to actually execute a service like binding, endpoint and service elements locations.
    - *Semantic web service descriptions* – if web services are additionally annotated in semantic manner using specific semantic web languages, like OWL-S (Semantic markup for Web Services – Ontology Web Language)[25] or WSML (Web Service Modelling Language)[26], it enables web service automatic discovery, invocation, composition and interoperation. On the other hand, more complicated structure of system is then designed and implemented, which sometimes overloads the real systems needs. It is usually needed especially for cases with dynamic workflows of business operations and incorporation of unknown (in design time) elements, like new devices and services.
- *Discoverability* – process of searching for services and retrieving information about them. In semantic web services different strategies and standards for discovery exist and is usually part of the semantic-based extensions. In basic case of non-annotated web services UDDI (Universal Description, Discovery and Integration)[27] standard is used. Implementers of the UDDI specifications can either be clients or servers, so called registries, which store various information about web services - business entity (publisher information), business service (descriptive information about service), binding template (technical information about service), tModel (generic container to summarize all technical information on the services).

In addition to several advantages, web services also have a few disadvantages, like performance (not as good as in distributed architectures that use binary protocols) and (for plain services) inexistence of infrastructure and quality of service features (such as security, transactions, etc). These issues can be solved by introducing additional WS* specifications [Juric et al., 2007], like WS-Security (addresses authentication and message-level security, and enables secure communication with web services), WS-Coordination (defines a coordination framework for web services and is the foundation for next specifications), WS-AtomicTransaction and WS-BusinessActivity (transaction specifications, support for distributed transactions, short duration by atomic, longer running transactions by BusinessActivity), WS-Reliable Messaging (support for reliable communication and message delivery between web services over various transport protocols), WS-Addressing (message coordination and routing), WS-Inspection (dynamic introspection of web service descriptions), WS-Policy (policies declarations and exchanges between collaborating web services), WS-Eventing (event model for asynchronous notification of interested parties for web services).

[24] WSDL 2.0 – http://www.w3.org/TR/wsdl20/

[25] OWL-S – http://www.w3.org/Submission/OWL-S/

[26] WSML – http://www.wsmo.org/wsml/wsml-syntax

[27] UDDI specifications – http://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm

3.1.2.6. Application servers

Application servers (APS) are software platforms, which are able to handle most of the interactions between clients and server tiers. They are not some very specific middleware technology, but provide a collection of already mentioned middleware services with management environment for deploying of business logic components. This environment is called *container* and (in majority of servers) can support web services, ORBs, MOM, transaction management, security, load balancing, and resource management. Due to these reason APS are suitable platform for integration. Many of the professional APS are able to specifically configure different middleware products.

APS is a combination of software technologies necessary to run applications, so they define the infrastructure of all applications developed and executed on them. Application servers can implement some custom platform, but standardized solutions are preferred now. The most important aspects of a platform are technical issues (software technologies, architecture of applications, portability, security, etc.), openness (possibility of influencing the development of the platform), interoperability, cost and maturity.

Java platform is widely used in application servers for support of J2EE functionality. There are several Java-based commercial products like Oracle WebLogic Server[28] or WebSphere Application Server[29] with many advanced middleware integration features. On the other hand, Java platform provides also good open-source solutions. One of the basic examples (with many standards used) is Glassfish Application Server[30]. Other suitable open-source solutions of application servers for J2EE are Apache Geronimo[31], JBoss[32], Sun GlassFish Enterprise Server (based on GlassFish APS), etc. There are also (so-called) light-weight application containers (not full application server functionality) – the most popular is open-source server Apache Tomcat[33].

Non-Java platforms also provide application servers. Zend platform provides an application server called Zend Server[34], which is used for running and managing PHP applications. It is a commercial product, but has also a version for free distribution (community edition). Open-source application servers are also available for other platforms, e.g. Base4[35] (for .NET applications) or Zope[36] (for Python). Due to the fact that non-Java APS are not formally specified within JSR (Java Specification Requests), their interoperability is low (in comparison to J2EE products). These problems are addressed by specifications of other technologies and standards like Business Application Programming Interface (BAPI, for SAP-based applications), Web Services Interoperability (WS-I)[37], and Java EE Connector Architecture (JCA)[38].

[28] Oracle WebLogic Server - http://www.oracle.com/technetwork/middleware/weblogic/overview/index.html

[29] WebSphere Application Server – http://www-01.ibm.com/software/webservers/appserv/was/

[30] Glassfish Application Server – https://glassfish.dev.java.net/

[31] Apache Geronimo – http://geronimo.apache.org/

[32] JBoss Application Server – http://labs.jboss.com/jbossas/

[33] Apache Tomcat – http://tomcat.apache.org/

[34] Zend Server – http://www.zend.com/products/server/

[35] Base4 server – http://base4.net/

[36] Zope application server – http://www.zope.org/

[37] WS-I – http://www.ws-i.org/

[38] JCA – http://java.sun.com/j2ee/connector/

3.1.2.7. Enterprise service buses

An Enterprise Service Bus (ESB) is a software infrastructure acting as an intermediary layer of middleware that addresses the extended requirements that usually cannot be fulfilled by web services, such as integration between web services and other middleware technologies and products, higher level of dependency, robustness, and security, management, and control of services and their communication.

Many vendors offer ESB products or products to set up or implement an ESB. They promise to ease intra- and inter-organisational connectivity, make possible smooth integration of legacy applications, enable to easily integrate various types of IT assets and so on. In general, ESB is one way to implement a company-wide or even intra-company SOA by providing a distributed middleware system for integrating enterprises IT assets. An ESB makes it possible to connect services implemented in different technologies (such as EJBs, messaging systems, web services, CORBA components, and legacy applications) in an easy way. An ESB can act as a mediator between different, often incompatible, protocols and middleware products.

Many of the integration problems (when combined) could not be solved satisfactory using already existing technologies like CORBA, APS, MOM or EAI approaches. ESB bridges the gap between traditional EAI solutions and MOM by combining the advantages of both integration approaches and adding even some more improvements. ESB can be configured rather than coded which allows a clear separation of application and integration logic. The central element of ESB is a *message bus* that is used as the communication medium and message broker between different components or applications.

ESB can be seen as a step beyond web services for SOA architectural framework, which promises a solution (as the primary goal of SOA) to align the business world with the IT world in a way that makes both more effective [High et al., 2005]. Utilisation of the ESB architectural pattern provides tremendous value when adopting a SOA.

Technically, ESB is a distributed infrastructure for enterprise integration and mainly consists of a set of services, based and interconnected with a reliable messaging bus sometimes viewed as a standard-based communication layer that enables services to be used across multiple communication protocols and data formats, which would include orchestration, adapters, management, and governance capabilities as part of their definition.

As a communication and integration layer an ESB should provide integration functionality through transformation, communication and routing. Service requesters and providers interact by exchanging messages. The core capabilities of an ESB are messaging, message transformation and message routing [La et al., 2007]. JMS is typically used as the message backbone, but any other message server implementation could be used (e.g. MSMQ, IBM MQ Series or TIBCO Rendezvous). Message transformations are provided within ESB for transparent exchange of messages between different systems, where XML is standard technology (with XSLT, XPath, XQuery) for implementation. Message routing is a process of routing messages between ESB service requesters and providers, where ESB basically supports static or adaptive routing.

More advanced functions of ESB could include different things like services and processes integration, integration adapters based on standards as JCA, management and monitoring, audit, logging, admin console, increased interoperability, QoS, security services, reliable message delivery, transaction management, etc.

ESB products are also designed in a standardized form. Java Business Integration (JBI) is a way of dealing with this. JBI provides a messaging and web services-based collaboration framework which provides standard interfaces for integrating third-party components and protocols to plug in. It defines a pluggable Java container, or execution environment, for integration solutions providing a messaging infrastructure for those components to interact with. In short, JBI acts as a container of containers, allowing various service engines and binding components to plug in and communicate with using a common messaging bus [Ten-Hove, 2006]. The JBI 1.0 (JSR 208)[39] and JBI 2.0 (JSR 312)[40] are industry-wide standards providing an open integration platform for Java and business applications. Both projects are being jointly developed through the Java Community Process (JCP) program by over 22 prominent vendors and individual developers of Integration and Java 2 Enterprise Edition (J2EE) technology, including Novel, Oracle (Sun), SAP AG, SeeBeyond, Sonic Software, TIBCO Software, JBoss, IONA and several more [Cover, 2004].

The following components are commonly referred to as the “core services” in JBI [Kinnumpurath, 2005]:

- Component Framework – enables the deployment of different types of components within the JBI runtime.
- Normalized Message Router – provides a standard mechanism of message interchange between services.
- Management framework – enables deployment, management and monitoring of components within the JBI runtime (based on JMX).

Basically these are the components that are defined in the JSR 208 specification and must be implemented by any JBI implementation. Several open-source JBI-based ESB implementations exist, like Open ESB[41], Petals ESB[42], Apache ServiceMix[43], FUSE ESB[44] (enterprise version of ServiceMix, compliant JSR 208), Bostech ChainBuilder ESB[45], Mule ESB[46] (provides interoperability with JBI containers), JBoss ESB[47]. Currently, Open ESB and Petals ESB are certified by the JBI/JSR 208. In addition, GlassFish open-source application server comes with the JBI runtime from the Open ESB project.

3.1.2.8. Integration from the view of BPM and workflows

A business process can be modelled as a workflow. Modelling can be done on different levels of details depending on the aim of the modelling (such as explanation, teaching and executing). Basically, a workflow model for support of BPM (Business Process Management) provides a mapping for key activities, decision point and work distribution. It consists of different modelling primitives,

[39] JBI 1.0 – http://jcp.org/aboutJava/communityprocess/final/jsr208/index.html

[40] JBI 2.0 page on JCR – http://jcp.org/en/jsr/detail?id=312

[41] Open ESB – https://open-esb.dev.java.net/

[42] Petals ESB – http://petals.ow2.org/

[43] Apache Service Mix – http://servicemix.apache.org/home.html

[44] FUSE ESB – http://fusesource.com/products/enterprise-servicemix/

[45] Bostech ChainBuilder ESB – http://sourceforge.net/projects/bostech-cbesb/

[46] Mule ESB – http://www.mulesoft.org/

[47] JBOSS ESB – http://jboss.org/jbossesb

most prominent being tasks. An executing instance of a workflow is called a process instance. During its execution, task instances are being created and executed. Completion of a task instance (as one of possible events) can initiate (conditionally or unconditionally) instantiation and execution of other tasks. Control flow between tasks is modelled by control-flow modelling primitives. Task instances are executed by resources to which they were allocated. This allocation can be modelled by resource modelling primitives. While being executed, tasks can communicate data elements between one another which can be modelled by data modelling primitives.

According to [Hollingsworth, 1995], Workflow Reference Model consists of five basic components: Workflow Enactment Service (creating, managing and executing workflow instances), Process Definition (tools to analyse, model, describe, and document a business process), Workflow Client Application (end user interaction), Invoked Application (software entities which are able to carry out task instances, currently mostly represented in the form of web services), Administration and Monitoring Tools (status monitoring, extracting metrics information, and management functions, security issues, etc.).

Development activities in the BPM area have already moved beyond a phase of ad-hoc vendor specific solutions and are governed by different specifications to produce portable solutions. More organisations and/or initiatives try to introduce their ideas about BPM, e.g. BPMI (Business Process Management Initiative), OASIS, OMG, W3C, WfMC (Workflow Management Coalition), etc.

To model business processes, different types of models can be used – the selection of a model type (with subsequent selection of a particular modelling technique) depends on the aim of modelling. When using a criterion who is an ultimate consumer of the model, two basic types of models can be distinguished: models for humans (understanding, communication and decision making) and models for machines (workflow engines, detailed process definitions that can be executed). A natural way of building process-oriented systems is to utilise both types, but it is still challenging due to different expressive power and syntactic restrictions.

In general, there are numerous modelling techniques in both categories for disposal, but each category has one dominant solution as a de facto standard: BPMN[48] (Business Process Modelling Notation) for visual modelling and BPEL[49] (Business Process Execution Language, short generalized name of WS-BPEL or BPEL4WS) for executable modelling.

BPMN can be viewed as an equivalent of UML in the area of process modelling. Its primary goal is to provide a notation that is readily understandable by different types of users, from business analysts to technical developers responsible for implementing the technology that will perform those processes, and finally, to business people who will manage and monitor those processes. Thus, BPMN creates a standardized bridge for the gap between the business process design and process implementation. The specification defines the notation and semantics of business process diagrams, like visual appearance of the BPMN graphical elements, semantics of the BPMN elements and possibility to exchange BPMN diagrams between conformant tools. The intent is to create a standard visual language that all process modellers can recognise and understand.

History of BPEL started with specification of BPEL4WS (Business Process Execution Language for Web Services), which combines older languages of consortium members. The proposal was revised, updated and submitted to OASIS as BPEL4WS V1.1. Technical committee for WS-BPEL then

[48] BPMN – http://www.bpmn.org/

[49] WS-BPEL Technical Committee – http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsbpel

prepared new specification for language called WS-BPEL[50] (Web Services Business Process Execution Language) V2.0.

WS-BPEL represents a language for specifying business process behaviour. It enables users to describe business processes in two ways – abstract and executable (both ways share constructs and have the same expressive power). Executable business processes model actual behaviour of a participant in a business interaction. Abstract business processes are partially specified processes that are not intended to be executed. An abstract process (must be explicitly declared as 'abstract') may hide some of the required concrete operational details to serve a descriptive role - it may be used to describe observable message exchange behaviour of each of the parties involved, without revealing their internal implementation.

The language allows describing behaviour of a business process based on interactions between the process and its partners. The interaction with each partner occurs exclusively through web service interfaces, and the structure of the relationship at the interface level is encapsulated in what is called a partnerLink. WS-BPEL also introduces systematic mechanisms for dealing with business exceptions and processing faults. Moreover, it introduces a mechanism to define how individual or composite activities within a unit of work are to be compensated in cases where exceptions occur.

The major building blocks of BPEL processes are activities. There are two types: structured activities can contain other activities and define the business logic between them. Basic activities only perform their intended purpose. It is possible to model providing and consuming web services, structure the process logic, define repetitive activities, parallel processing, manipulations with data, and many additional advanced concepts.

The standardized WS-BPEL 2.0 differs from BPEL4WS 1.1 in several ways, like new activity types, variable initialization, XPath access to variable data in a simplified manner, XSLT for variable transformation, clarification of abstract processes, etc. Some of the changes are quite significant and are a source of incompatibility between the two languages (e.g. syntax changes, modifications of semantics of existing constructs), so the migration is not trivial.

To overcome exclusion of human tasks in WS-BPEL, BPEL4People specifications were defined, with latest proposal defined by OASIS Technical Committee for a new WS-BPEL Extension for People (BPEL4People)[51] specification. The BPEL4People extension is defined in a way that it is layered on top of WS-BPEL. It introduces a set of elements which extend the standard WS-BPEL elements and introduce the modelling of human interactions.

Since BPMN is envisaged as a visualisation of processes which can be made executable using BPEL, the BPMN standard deals with mapping to BPEL4WS. Suggested mapping is provided for business process diagrams, business processes, common flow objects, events, activities, gateways, sequence and exception flows (some objects are not mapped e.g. pools and lanes). As can be seen from the analysis how those two standards cover workflow patterns, BPMN is able to support more patterns than BPEL. Intuitively, not all attempts to map a graphical BPMN-based model into a XML-like BPEL-based model will end successfully. There are some graphical models which cannot be mapped into executable BPEL. The fundamental reason for this is that BPEL imposes far more syntactic restrictions than BPMN.

Several works can be found dedicated to translating BPMN models into BPEL process definitions for subsequent refinement (data manipulation, web service binding, etc.), e.g. [Ouyang, 2006] shows a

[50] WS-BPEL 2.0 – http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html

[51] BPEL4PEOPLE – http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=bpel4people

technique enabling to translate every model build on a core BPMN subset using several translation techniques that can be combined together.

3.1.2.9. Content and presentation integration – portals and content repositories

Integration of presentation layer and content are important aspects of integration infrastructure. In spite of their differences from user point of view, they often have very tight cooperation within integrated system. Content integration based on content repositories is usually important back-end solution for content sharing, which is then shown in presentation layer and is mostly based on some personalised web portal. Portals can be characterized in different words, but there is also an exact technical solution based on the specification of Java Portlet Specification JSR 168[52]. According to this specification, "a portal is a web-based application that – commonly – provides personalization, single-sign-on, content aggregation from different sources and hosts on the presentation layer of information systems. Aggregation is the action of integrating content from different sources within a webpage. A portal may have sophisticated personalization features to provide customized content to users. Portal pages may have different sets of portlets creating content for different users."

Different types of portals can be distinguished. One differentiation is vertical (specialise in detail about one specific subject) and horizontal portals (broad range of information provided). Another distinction can be for open and closed portals. Most interesting is to divide portals into:

- *Process-oriented business portal* – closed user group with access to (automatable) business processes in a consistent fashion.
- *Application-oriented business portal* – aggregates selected business applications and their respective data sets into the user interface of the application.
- *Consumer portals* – horizontal portals incorporate different sources of information into one consistent user interface.

From an application developer's perspective, portals based on Portlet 1.0 consist of several independent web applications, called portlets, which are combined together into one uniform user interface, running under a Java application server within portlet container. The Java portlet specification JSR 168 defines a standard for individual portlets, thus enabling platform independence of portlets, aiding usage across different application servers and thereby guaranteeing a high degree of transportability. The portlet specification JSR 168, released in 2003, defines a set of 12 classes and 14 interfaces, which assure compatibility between a portlet container and the portlet itself. One drawback of JSR 168 is that individual portlet instances running in one portal cannot communicate with each other. A first draft of the Portlet API 2.0[53] which has been released in 2006 and has subsequently been published as JSR 286 tackles this problem. Its main focus is to enable communication between individual portlets, the so-called inter-portlet communication (IPC). A single portlet is to be provided with the possibility "to send and retrieve events and perform state changes or send further events as a result of processing an event".

Another standard comes from web services world. The main focus of WSRP (Web Services for Remote Portlets)[54] is on interactive, presentation-oriented services. Mainly, the WSRP standard comprises execution interface for WSRP services (using WSDL), rules for interaction with WSRP

[52] Java Portlet Specification 1.0, JSR 168 – http://jcp.org/en/jsr/detail?id=168

[53] Java Portlet Specification 2.0, JSR 268 – http://jcp.org/en/jsr/detail?id=286

[54] WSRP OASIS Technical Committee – http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrp

services (which cannot be expressed using WSDL) and rules for the structure of data to be created by WSDL services and sent to other clients. Communication using WSRP involves (as actors) portlet itself (offering presentation-oriented services), producer (embedding one or more portlets and offering these as web services) and consumer (making use of services offered by one or more producers). SOAP is used for communication. WSRP standard in 1.0 also did not specify mechanisms of communication between individual portlets, but current version called WSRP 2.0 already supports this feature.

There are several available open-source portal products, like Apache Cocoon[55], Apache Pluto[56] (reference implementation of JSR168 and JSR268), Jetspeed 2 Enterprise Portal[57], GateIn[58] (combination of eXo and JBossPortal, from which GateIn evolved), Sun Java System Portal Server – OpenPortal[59] (server under Sun GlassFish Web Space Server), Liferay[60], etc.

The purpose of the Java Content Repository API (JCR) is to ease access of Java applications to digital content of any kind, where content is stored together with metadata used in CMS. JCR 1.0[61] was released in 2005 under JSR 170, followed by JCR 2.0[62] under JSR 283. The main goal is the unification of different content management applications. Formerly, every content management application used to store content in a (frequently proprietary and thus inaccessible to other applications) content repository. This repository usually offers services necessary to facilitate document management like, for example, versioning of one document. In order to enable interoperability between document management systems, JCR introduces a unified API that allows accessing any compliant repository in a vendor- or implementation-neutral fashion. Apache Jackrabbit[63] is well-known open-source JCR implementation, one of the new ones is ModeShape[64].

Content integration is basic ECMS (Enterprise Content Management Systems) feature. One of the problems in ECMS field is interoperability of ECMS products. Content Management Interoperability Services (CMIS)[65] is a specification for improving interoperability between ECMS products. CMIS provides interface, which is expected for good interoperability of the ECMS software. One of the standardization leaders in ECMS system technologies is an open-source solution called Alfresco[66], which also has strong collaborative components and portal-based features in its Alfresco Share front-end interface.

3.1.2.10. Other types of technologies

Another type of integration technologies is called Transaction Processing Monitors (TPM). This solution is based on the concept of transactions and therefore it is important for mission-critical

[55] Apache Cocoon – http://cocoon.apache.org/

[56] Apache Pluto – http://portals.apache.org/pluto/

[57] Jetspeed 2 – http://portals.apache.org/jetspeed-2/

[58] GateIn portal – http://www.jboss.org/gatein

[59] OpenPortal, Sun Java System Portal Server within Glassfish server – https://portal.dev.java.net/

[60] Liferay – http://www.liferay.com/

[61] JCR 1.0 – http://www.jcp.org/en/jsr/detail?id=170

[62] JCR 2.0 – http://jcp.org/en/jsr/detail?id=283

[63] Apache Jackrabbit – http://jackrabbit.apache.org/

[64] ModeShape – http://www.jboss.org/modeshape

[65] CMIS – http://docs.oasis-open.org/cmis/CMIS/v1.0/os/cmis-spec-v1.0.pdf

[66] Alfresco – http://www.alfresco.com/

applications and represents the first generation of application servers. TPM have several tasks: monitor and coordinate transactions among different resources, providing performance management (load balancing, pooling) and security services. Due to these facts TPM are predecessors of application servers. They have been traditionally applied in legacy information systems using procedural models, RPC, APIs. TPM are proprietary products, which make migration from one product to another very difficult and not interesting for our purposes.

3.1.3. Evaluation of integration technologies

In this section, evaluation of previously presented integration technologies according to the current version of requirements from D1.1 [Bicking et al., 2010] and a basic vision of the future integrated system is described. The purpose of this evaluation is to see which technologies (or groups of technologies) are suitable to fulfil our needs for the implementation of the prospective OCOPOMO platform.

In D1.1 there are presented several groups of requirements which are directly related to integration of our system. Table 4 will provide for each group the following information: Related requirements (Name with ID) in *Reqs* column, Type of requirement (F - functional, NF - non-functional) as *Type*, and *Evaluation* comments related to a connection of requirements to some concrete technologies (whether there are some consequences from a particular selection). The one basic aspect of the evaluation is that in OCOPOMO we prefer a simple solution, especially where some open-source tools could be reused together with a subset of their technologies and functionalities. More detailed information on particular requirements can be found in already mentioned D1.1 deliverable [Bicking et al., 2010].

Requirements group	**Reqs**	**Type**	**Evaluation**
User and profile management	Password reminder (I-F-I1) Removing profile (I-F-I2) User registration (I-F-I4) User profile (I-F-I5) All personal preferences in one place (I-36) *	F F F F F	- usual requirements on web portals - should be reused within selected integration solution Suitable integration technologies: Content and presentation integration within portals based on Portlet/Application Server Containers. Portlet technology is not mandatory, but it is probably good solution to stay within one application server, which will allow most of the functionality, or it is easy for extension. Other useful technologies: Database access technologies usually realized within portal solution are used for profile persistence.
Security	Login (I-F-I3) Privacy (I-NF-6) Authorization (I-NF-5) Authentication (I-NF-4)	F NF NF NF	- in our case a simple solution without a specific security extension module is expected (reused security model of selected software tools, etc.) Suitable integration technologies:

	Integrity (I-NF-10)	NF	Reuse of security model from selected integration technology should be enough for our purposes. According to character of requirements, again content and presentation integration within portal running in application server seems to be a solid solution with their security model, which can be shared within application container.
Graphical user interface	Multilingual interface (I-35) Personalise overview (I-F-16) ICT toolbox functionality provided through one portal-based interface (I-1)	NF F F	- integration of views from different tools - personalized dynamic content Suitable integration technologies: Mostly related to GUI of web portals and sharing of content – again content and presentation integration technologies are most important.
Efficiency	Response time (I-NF-3)	NF	- hard to have some pre-selection for this requirement, maybe two aspects are important: - less pieces of software for integration is probably better - integration of most functionality within one software and only smaller addition of several others is better
Usability and accessibility	Usability (I-NF-1) Look and feel (I-NF-8) Help and assistance (I-NF-11) Accessibility (I-NF-2) Operational (I-NF-7)	F NF NF NF NF	- in this case we have again quite presentation-related requirements Suitable integration technologies: Similarly to GUI requirements – content and presentation integration technologies.

Table 4 Evaluation of D1.1 user requirements classified as integration-related requirements (all the requirements in the table default to 'must-have' priority, * indicates 'nice-to-have' priority)

If we want to summarize the previous evaluation, we have to say that (logically according to the fact that the presented user requirements are more content and presentation related) content and presentation integration technologies are very important and this layer has to be clearly defined and prepared for the implementation.

According to current analysis, content and presentation technologies (portlet or non-portlet-based portals, content integration based on the content management technologies, etc.) combined within application servers (which supports most of the "classic" integration techniques using standards, like EJB, SOA, etc.), which have persistence and security solved using some standard database access technologies and own security models, are fully suitable to give integration of any tools real platform. So, for the summary, if we will use such combination of integration technologies, we are able to fulfil:

- Data-level integration – using database access technologies, supported by the content integration (e.g. content repository) for advanced CMS functionality
- Application integration – using application server and its container, any technology which is supported within container can be helpful for tight integration of specific parts of the platform (mostly preferred is API sharing)

- Presentation integration – most important layer in this evaluation, should be better to use some existing portal solution and add some additional parts of other software or implement new one, where it is needed – content and presentation integration technologies are fundamental here.
- Business integration – mostly important for previous case as a formal modelling step for supporting content sharing using specific process and document workflows. We will probably do not need any other technologies, since the previously mentioned ones usually have workflow support available these days.

3.2. E-PARTICIPATION TOOLS AND TECHNOLOGIES

Relevant studies have recently evidenced an increasing activity in the field of e-participation in Europe [Aichholzer and Allhutter, 2009], [Panopoulou et al., 2009], [Scherer et al., 2008], [Tambouris et al., 2008], [Tambouris et al., 2007]. Due to the breadth of the field, a number of distinct tools are used to support the different e-participation areas[67] (see for example [Tambouris et al., 2007], [Thorleifsdottir and Wimmer, 2006], [Wimmer, 2007]).

3.2.1. Description of available alternatives

Today, some e-participation offerings are implemented in a very simple manner using standard software available, such as on-line forms or discussion forums or are based on content management systems (CMS) and include further functionalities. But many of the more comprehensive offerings use specialized software tools for e-participation. As [Albrecht et al., 2008] describes, Wikipedia is an example of a technically relatively simple system, which “shows that e-participation can be carried out with very good success using simple tools” (p. 84). They underline that tools used is less important than the concept and methodological design of participation offerings, what was already described in D1.1 [Bicking et al., 2010]. The integration of technologies and suitable methods is a critical success factor for e-participation [Thorleifsdottir and Wimmer, 2006].

In order to define criteria for selecting appropriate tools for certain e-participation areas, Table 5 can be used as a first base. On one hand, the table shows which tool categories are used extensively (black cells) or in a supportive way (grey cells) in different e-participation areas [Scherer et al., 2011]. On the other hand, the last row of the table visualises, how often which tool is used in 13 e-participation projects co-funded from the European Commission (based on [Bicking and Wimmer, 2009], [Charalabidis et al.,2009]). E-participation areas and tools relevant in OCOPOMO [Bicking et al., 2010] are marked underlined and bold. The e-participation tools named in the table and their usage in e-participation are elaborated in DEMO-net deliverable 5.1: Report on current ICTs to enable Participation [Thorleifsdottir and Wimmer, 2006].

[67] E-participation areas are specific and distinct sectors of the democratic process, which are defined by the context and the scope of electronic participation [Thorleifsdottir and Wimmer, 2006], [Westholm and Wimmer, 2007].

Tools **Area**	**Content management systems**	**Discussion forums**	**Weblogs**	E-petition systems	E-voting systems	**E-consultation**	E-surveys & e-polls	**Online meetings and chats**	Serious games	**Community systems**	GIS and Map based tools	**Combined collaboration systems**
Information provision	■		■						▒			
E-consultation	■	■	▒			■	▒	■			▒	
E-petitioning		▒		■								
E-voting					■							▒
E-surveying & E-polling						▒	■					
E-lobbying	■	▒	■						▒			
E-electioneering	■	▒	■			■	■			■		■
E-collaboration		■	▒				▒	■	▒	■	▒	■
E-empowering		■	▒	▒	▒		▒	■	▒	■	▒	■
Usage	***5***	***6***	***3***	***0***	***0***	***0***	***3***	***3***	***1***	***2***	***2***	***2***

Table 5 Different electronic tool categories used in different e-participation areas ('extensive use' in black, 'supportive use' in grey) (based on [Scherer et al., 2011])

Resulting from requirements analysis in Work Package 1 [Bicking et al., 2010] and tool categories proposed in Table 5, relevant software types for the OCOPOMO platform are (descriptions of these types are available in [Thorleifsdottir and Wimmer, 2006]):

- Content management systems
- Discussion forums
- Weblogs
- E-consultation
- On-line meetings and chats
- Community systems
- E-surveys and e-polls
- Wiki

These functionalities as well as related information need to be on-line available and integrated into the OCOPOMO platform. Participation facilities must be available for users without the need to install any software. Information needs to be linked with participation offerings and vice versa.

In order to integrate the functionalities into the OCOPOMO platform, two possible approaches exist. On one side, different existent participation and collaboration tools could be selected and integrated into one platform. But in terms of usability, the use of different participation features must be well-considered to not overload users [Scherer et al., 2009b]. It is of course easier for users, if features provided have a similar look and feel and are integrated in one environment. Therefore the second option is to use a web CMS, which already provides most of these tools and integrates them into one platform. If further functionalities are needed, these can be integrated as a plug-in into the CMS. Currently, considering project resources, we prefer the latter approach.

As many content management systems provide these features, they will be analysed in this regard in the section 3.2.1.1. Particular designed solutions for e-participation are investigated in section 3.2.1.2. As wiki functionalities have been required in the requirements analysis, wiki software will be mentioned in section 3.2.1.3.

3.2.1.1. Content Management Systems

The "Open Source CMS Market Share" in 2009 analysed the brand strength and market share of 20 open source web content management systems [cms, 2009]. As such, it provides important information relevant to selecting a CMS. But as the study states, it should not be read as a final judgment on the feature quality, stability, or a particular system's suitability. Rather it aims at providing a body of useful data which enable to make a more informed decision about which product is the best fit. The 20 systems covered in this report have been assessed on variety of metrics related to Rate of Adoption and Brand Strength. The analysis looked at a broad range of indicators – both direct and indirect – with the goal of synthesizing trends and patterns. Conclusion of the study: the open source CMS market is dominated by WordPress, Joomla! and Drupal - the same result was found even in the last year's study. The fact that all three systems are programmed in PHP is typical for this market: even if the study takes into account different .NET, Java, and Python systems, PHP is still the dominant language for open-source CMS. Alfresco is on the ascending branch. On the other hand, Plone and Xoops recorded overall declining values. Typo3 was ranked in the middle of the field. CMSs, which are not open source, e.g. the Microsoft Office SharePoint Server, have been not considered. As a result from this investigation, seven CMS have been selected to analyse them for their support of different functionalities usable in the OCOPOMO platform:

- *Alfresco / Alfresco Share*[68] is a leading Java-based open source enterprise content management system for documents, web, records, and collaborative content development. Alfresco has strong support for integration with enterprise technologies (e.g. SharePoint) and desktop office applications using the open content management standards like CMIS[69] (Content Management Interoperability Services, OASIS standard).
- *Drupal*[70] is a free open-source platform and content management system for building dynamic web sites. It offers a range of features and services including user administration, publishing workflow, discussion capabilities, news aggregation, metadata functionalities using controlled vocabularies and XML publishing for content sharing purposes. Equipped with a powerful blend of features and configurability, Drupal can support a diverse range of web projects ranging from personal weblogs to large community-driven sites. In general, Drupal focuses on communities and collaboration.
- *Joomla*[71] enables to build Web sites and powerful on-line applications. Many aspects, including its ease-of-use and extensibility, have made Joomla the most popular Web site software available.
- *Plone*[72] is a ready-to-run content management system that is built on the free Zope application server. It is free and open source. Plone is easy to set up, flexible, and provides users with a

[68] http://www.alfresco.com

[69] http://docs.oasis-open.org/cmis/CMIS/v1.0/os/cmis-spec-v1.0.html

[70] http://drupal.org

[71] http://www.joomla.org

[72] http://www.plone.org

system for managing web content for project groups, communities, web sites, extranets and intranets. The egosta portal[73] for participation of stakeholders in e-government projects in Austria uses Plone [Ventzke et al., 2010]. Plone has been selected here although it has recorded overall declining values because the OCOPOMO website bases on this CMS.

- *TYPO3*[74] is a web content management framework, based on PHP and MySQL. It is a free open source content management system for enterprise purposes on the web and in intranets. Typo3 is a very complex content management system. It is a server-side platform-independent application that can be used with virtually every browser available. TYPO3 is database-driven and scales easily to deliver web pages and embedded formats in an enterprise content providing environment. A number of extensions are available in the extension repository, which only consists of freely available extensions. Typo 3 has been selected because it is a very comprehensive CMS.
- *WordPress*[75] is an open source CMS. It has grown from a pure blogging focus into a full-fledged content management system. The default system is focused on blogging, but a large number of open source plugins are available to extend the functionality.
- *XOOPS*[76] is an extensible, object oriented, dynamic web content management system written in PHP. XOOPS can be used as a tool for developing small to large dynamic community websites, intra company portals, corporate portals, weblogs, etc. A number of modules are available for the environment, but there is only a small developer group. It has been selected as XOOPS has been designed to support communities in particular.

Table 6 shows an overview of the seven CMS with additional information about latest version, license, application server, operating system, database, programming language, interfaces and web server.

	CMS						
	Alfresco	**Drupal**	**Joomla**	**Plone**	**TYPO3**	**WordPress**	**XOOPS**
Latest version	3.3	6.19	1.5	3.3.5	4.4	3.0.1	2.4.5
License	GNU General Public License v2	GNU General Public License v2	GNU General Public License v2	GNU General Public License v2 or later	GNU General Public License v2/v3 (upcoming version 5) or later	GNU General Public License v2	GNU General Public License v2
Application server	J2EE	Apache	Common Gateway Interface	Zope	Apache, ISS	Apache	Apache
Operating system	Platform Independent	Platform Independent	Platform Independent	Platform Independent	Unix (e.g. Linux), Windows or Mac	Platform Independent	Platform Independent

[73] http://www.egosta.at

[74] http://www.typo3.org

[75] http://www.wordpress.org

[76] http://www.xoops.org

Database	MySQL, PostgreSQL	MySQL, PostgreSQL	MySQL	ZopeDB (object oriented)	MySQL, Oracle, Postgres and others	MySQL	MySQL
Programming language	Java	PHP	PHP	Python	PHP	PHP	PHP
Interfaces	Standards-based JSR-168 and REST-based integration, XHTML compliant, RSS, FTP support, WAI compliant	XML-RPC, blogapi's, with additional modules XML, CSV, diverse HTML variants, PDF, XHTML compliant, RSS, iCal, , WAI compliance limited	phpMyAdmin, MySQL and SQL statements, iCal, RSS, FTP support	XHTML compliant, iCal, RSS, FTP support, WAI compliant	Interfaces for all common interchange formats, XHTML compliant, iCal, RSS, FTP support, WAI compliance	RSS, iCal, , WAI compliance limited, XHTML compliant	iCal, RSS, FTP support
Web server	Any	Apache	Apache	Apache	Apache, IIS	Apache	Apache

Table 6 Basic characteristics of the selected CMSs

3.2.1.2. E-participation platforms

Besides these CMS, which are more or less customisable in order to be usable for e-participation, a number of specialized software tools exist for e-participation. In this regard, the following three tools can be mentioned.

Gov2Demos[77] is an open source, customizable, informative and collaborative e-participation platform that serves as a proof of concept of how ICT can facilitate communication, knowledge sharing, and modernization of government services. Gov2Demos is based on Joomla and therefore supports all Joomla functionalities. Beyond, Gov2Demos is further customised for e-participation [Koulolias et al., 2006]. Gov2Demos has e.g. been used for the VoicE/VoiceS platform[78]. In the VoiceS project, it has been extended for a semantic web search engine and a range of other functionalities [Scherer et al., 2009a].

The Discourse Machine[79] is a comprehensive software system which supports the management of online discussions. Different tools, including wikis, weblogs and interactive graphics, can be combined. Since the range of functions and the user interface can be adapted to the respective requirements, the system can be set up to meet the demands of special target groups [Albrecht et al., 2008]. The

[77] http://www.gov2u.org

[78] http://www.bw-voice.eu (German) or http://www.voice.gva.es (Spanish)

[79] http://www.discourse-machine.de

Discourse Machine is not open source but there are license models, which allow it to customize and extend the software.

ICELE[80] offers a variety of free and low-cost electronic tools, including a community website and portal solution that is currently being piloted.

These free software tools are not further analysed for the following reasons:

- As Gov2Demos bases on Joomla, in a first step it is sufficient to analyse this system for its base functionalities. In the case that Joomla is selected as one of the tools which come into consideration, Gov2Demos could be further tested.
- The Discourse Machine is not further analyzed because it is not open source software.
- ICELE is not further analyzed because the portal solution is currently in a pilot stage only.

3.2.1.3. Wiki software

Wikis are web applications that allow users to add, remove, edit and change content collectively. Users can change the content of pages and format them with “a very simple tagging language” [oecd, 2007]. Some wikis have strict moderation policies; others are less restricted, dependent upon the user group. It is generally the case that a clear statement of the rules of engagement makes for a more effective collaborative experience. The fundamental concept is that a large number of users read and edit the content, potentially enriching it and correcting mistakes [oecd, 2007].

Wiki software (wiki engine, wiki application) is a type of collaborative software that runs a wiki. The content, including all current and previous revisions, is usually stored in either a file system or a database. Some wiki software, e.g. MediaWiki[81], stores data in a database. Other wiki software, e.g. PMWiki[82], stores data in flat files. The former is more scalable.

There are a number of factors, which are important to the decision for wiki software as e.g. costs, complexity, control, clarity, common technical framework, and features. As in academia, the fact that most wiki software lacks tools such as instant messaging or link checking [Schwartz et al., 2004], can be a limitation in their usefulness for e-participation. Therefore, the focus in this analysis is on CMS, which can also be used in order to provide wiki-like functionalities.

3.2.2. Definition of criteria for selecting tools to incorporate into ICT toolbox

In order to define criteria for evaluating tools to incorporate into ICT toolbox, Table 7 shows and describes in detail the criteria which are used in the presented state-of-the-art analysis:

[80] http://www.icele.org

[81] http://www.mediawiki.org

[82] http://www.pmwiki.org

Functionality	Description
CMS (T-5)[83]	Key functionalities of a CMS
Web publishing, retrieval & browsing	Web publishing means all functionalities for creating and publishing documents. At least the WCMS has to support the document types *.doc, *.html, *.jpg and *.gif. It has also to support functionalities for creating, uploading, editing, searching and downloading these documents. Additionally, functionalities as comment and rate content can be supported.
Multilingualism (I-35)	Support of different languages.
Workflow engine (new requirement)	Workflow engine to manage e.g. publication and review workflows.
Layout/ Templating (T-5)	Using templates to manage layout.
Content/ WYSIWYG (new requirement)	What you see is what you get editor
User and rights management (T-5)	User and rights management is about defining and managing users and user rights.
Single sign-on (T-37, I-1)	Standards provided to support single sign-on.
Versioning (T-5)	Versioning of content so that it is possible to undo changes.
Customisable content types (I-14 to show scenarios)	Support of own customisable content types
Login – with e-mail or user name and password (I-F-I3)	After the initial registration, members can login each time they wish to access the site by providing their user name or email and password.
Remove profile (I-F-I2)	If a registered user wants to delete his/her profile and stop being a registered member, he/she must/can do this in the system. If the user is logged-in he or she needs to press the “remove my account” button and confirm this decision afterwards to remove the profile.
Community systems	
Comment content (T-25)	Authorized users (e.g. facilitator in case of the scenario generation) can decide whether the content in the system can be commented upon. Commenting should have always the same style, does not matter what is commented. Users are able to comment most of the sources within the system.
Rating content (T-C2)	Users are able to rate/vote for interesting news entries. Rating/Polling is an easy to use functionality to initialize first participative behaviour and interest with the topics.
On-line meetings and chats (T-4)	This functionality requires the possibility to integrate a chat program. Further on, there has to be the possibility to hold a video conference.
Personalised profiles (I-F-I6)	Personalised profiles with information about users.
Discussion (T-1, T-1-1 – T-1-5)	Discussion is about providing forum functionalities. Therefore we have to differentiate between moderated and not moderated forums. The discussion forum needs to be customisable in order to support needed functionalities.
Moderated and non-moderated discussions (T-12)	
Visibility of discussions for certain user roles (T-1-4)	
Multiple instances of a forum (T-	

[83] Identifiers in parentheses represent IDs of user requirements, identified in D1.1 [Bicking et al., 2010]

	1-1)	
	Entries should be organised in threads (T-1-2)	
	Possibility to order entries in chronological order and for topics (T-1-3)	
	Rating of contributions and contributors (analysis of discussions based on a relevance feedback) (T-14, T-C2)	
Mail		Possibility to send e-mails with the system.
Calendar (T-28)		The calendar should provide different views, like a daily, weekly and yearly diagram. There should also be an import and export function. This function will afford the exchange of appointments with local calendar programs like outlook.
Notification		For this, it should be checked, which options the WCMS affords to reach a user by mail or RSS feed if contents changed. Another application of those notifications can be the reminder of important appointments. User can choose how often he wants to get notifications.
	RSS feed (T-30)	
	E-Mail (T-34)	
Polling (T-7 – T-11, I-10)		To feature this functionality, the WCMS must be able to integrate a survey which can be answered by click, by a free answer or by choosing a given answer.
	open forms (authorized access, open/close polls) (T-7)	
	participation of users in polls – one vote per person (T-8)	
	possibility to modify the answers provided (versioning) (T-9)	
	different types of questions & answers (T-10)	
	(graphical) presentation of the results (T-11)	
Blogging		Integration of a blog into the CMS.
Wiki (T-39)		Wiki should be a collection of websites, which cannot only be read by the users but also be edited by them. It should also afford some users to work together on texts and definitions. Because of this, it is important to check if the WCMS supports the initialization of Wiki. Another option would be to extend the CMS with a wiki like functionality that allows creating scenarios.
Newsletter (T30)		Functionality to send newsletters to registered users.

Table 7 Criteria for selecting e-participation tools

Another important criterion is the possible integration into the ICT toolbox. It is also very important to know how it is possible to re-use any software which can be found as a solid alternative. All introduced CMS are open source and published under the GPL 2 or higher. Regarding standards used for interoperability issues (content integration, data exchange, etc.), it needs to be first concluded which standards can and need to be supported based on the needs of the other parts of the ICT toolbox (relevant standards are enumerated in Appendix A). Therefore the standards are not analysed further in this part. A base overview of interfaces enabling to access functionality of the tools is given in Table 6.

3.2.3. Evaluation of tools

Evaluating and comparing Web-CMSs is a difficult task as they mostly differ in minor details [Mintert, 2010]. To give a first overview, Appendix B shows a comparison of CMS. In addition, the seven chosen CMS are evaluated against the criteria for selecting tools introduced in the previous section (see Table 8).

Functionality	Alfresco	Drupal	Joomla	Plone	TYPO3	Word-Press	XOOPS
CMS							
Publishing, Retrieval & Browsing	yes	yes	yes	yes	yes, google-like search	yes	yes
Multilingualism	yes	yes	yes	yes	yes	yes	yes
Workflow engine	yes	limited	simple workflow system	yes	limited	no, simple add-ons	no
Layout/ Templating	form authoring using XML schemas, automatic user interface rendering based on XForms standard, automatic creation of multiple formats for multiple channels	Themes compliant with XHTML standard, barrier-free	PHP-Templates with JavaScript/ CSS/ HTML	Skinable interface.	TypoScript, TemplaVoila	yes	Theme-based skinable interface
Content/ WYSIWYG	yes, HTML editor	yes, common editors like tinyMCE instance, HTML, Area, FCKEditor, text entry via XML-RPC	yes, TinyMCE, JoomlaFCK, TMEdit, JCE, integrated image management	yes, FCKEditor	yes, rich text editor, alternatively others	Different add-ons e.g. based on widgEdit	yes, FCKeditor
User and rights	Security and user	definable, finely	simple pulley	definable, finely	definable, finely	yes, simple	yes, enables administrat

Functionality	Alfresco	Drupal	Joomla	Plone	TYPO3	Word-Press	XOOPS
management	management with users, groups and roles,	granulated	system with pre-defined groups and rights	granulated	granulated	integrated system, extendable with add-ons	ors to set permissions by group
Single sign-on	Single sign-on through NTLM or LDAP	between Drupal systems, OpenID	Kerberos Single sign-on, LDAP (but only intern recommended), OpenID (beta)	LDAP, OpenID	Single Sign-On Framework solution, OpenID with add-in	OpenID	OpenID
Versioning	Simply rollout a new site with automatic site versioning	integrated version management system	simple version control app,	versioning add-on for content items	automatic site versioning	yes with add-on	yes, versioning module for articles
Customisable content types	yes	yes	limited	yes	yes	yes	yes
Login – with e-mail or user name and password	yes	yes	yes	only with user name or with e-mail	yes (beta)	only with user name or with e-mail	solutions available
Remove Profile	yes	yes	yes	yes	yes	yes	yes
Community systems							
Comment content	yes	yes	yes	yes	yes	yes	yes
Rating content	no	yes	yes (like or dislike button)	no	no	yes	no
On-line meetings and chats	yes (beta)	yes	yes	yes	yes	yes	yes
Personalised profiles	yes	yes	yes	yes	yes	yes	yes
Discussion							
Moderated and non-moderated discussions	yes (collaboration product)	yes	yes	yes	yes	yes (bbPress)	yes
Visibility of discussions for certain user roles	yes (collaboration product)	yes	yes	yes	yes	yes (bbPress)	yes
Multiple instances of a forum (T-1-1)	yes	yes	yes	yes	yes	yes	yes

Functionality	Alfresco	Drupal	Joomla	Plone	TYPO3	Word-Press	XOOPS
Entries should be organised in threads	yes (collaboration product)	yes	yes	yes	yes	yes (bbPress)	yes
Possibility to order entries in chronological order and for topics	no	no	no	no	no	no	no
Rating of contributions and contributors	no	no	no	no	no	no	no
Mail	yes	yes	yes	yes	yes	yes	yes
Calendar	yes	yes	yes	yes	yes	yes	yes
Notification							
RSS feed	yes	yes	yes	yes	yes	yes	yes
E-Mail	yes	yes	yes	yes	yes	yes	yes
Polling	no	yes	yes	yes	yes	yes	yes
open forms (authorised access, open/close polls)	no	yes	yes	yes	yes	yes	yes
participation of users in polls – one vote per person	no	yes	yes	yes	yes	yes	yes
possibility to modify the answers provided (versioning)	no	no	no	yes	yes	no	no
different types of questions & answers	no	yes	yes	yes	yes	yes	yes
(graphical) presentation of the results	no	yes	yes	yes	yes	yes	yes
Blogging	yes	yes	yes	yes	yes	yes	yes
Wiki	yes	yes	yes	yes	yes	yes	yes
Newsletter	yes	yes	yes	yes	yes	yes	yes
Total							
yes	23	29	27	29	28	28	27
yes with limitations	1	1	3	1	2	2	1
no	9	3	3	3	3	3	5

Table 8 CMS support for different functionalities

Table 8 shows for each aforementioned CMS, which functionalities are supported either out of the box or as add-on and which are not supported. The analysis does not make any declarations about effort needed to implement or integrate plug-ins or add-ons necessary. In addition, in this step, it is not possible to rate, how good and usable existing implementations are. The colours green, orange and red give only indications about the availability of features.

The results are based upon desk research (e.g. studying the product websites and other web references). Most features are provided by add-ins. This makes it difficult to estimate if chosen add-ins work together smoothly.

Most of analysed functionalities are supported by all analysed CMS; either by integration or as add-on. As all selected CMS are open source, functionality, which is not provided until now, could be implemented. On major difference between chosen CMS lays in the support of workflows and versioning (see also [Mintert, 2010]). As this is not a functionality, which can easily be implemented, only CMS that provide good support for versioning and workflows should be further considered if the scenario building functionality should not be implemented with a wiki software tools.

3.3. SCENARIO GENERATION AND ANALYSIS TOOLS AND TECHNOLOGIES

Scenarios have been developed by the RAND Corporation in the early fifties when Herman Kahn worked out strategically studies on military issues on behalf of the American government [Von Reibnitz, 1987]. Scenario building received a significant boost and was copied by well-known organisations such as Shell and Global Business Networks later. The simple 'what if' exercises performed by national armies turned into fully-fledged future research methods [May, 1996]. In the 1960s and 1970s Gibson [Gibson, 1996] found that a general sense of certainty existed about where we were going and how to get there. However, the lesson learned is that nobody can just drive to the future on cruise control. During the twentieth century, a more down-to-earth approach was forced to look into the future. Consequently, the scenario method became also more mature (e.g. [Johnson et al., 2002] and [May, 1996]).

The purpose of the scenarios is to stimulate different internally consistent alternatives of either as-is or to-be situations and its settings within a specific problem scope. Scenario building provides the opportunity to collect information about a system of a certain problem scope, which is difficult to access. Scenarios help to identify the framework conditions of the system in order to allow better handling complexity and related uncertainty, and therewith better predictions for evolution. Based on the insights from such alternatives, concerted and focused models can be derived that describe the system and the behaviour of its elements. Scenarios focus on the identification and description of impact factors as well as cause and effect interdependencies [Straeter, 1988].

Geschka and Hammer classified scenarios on the basis of the scenario building process as follows [Geschka and Hammer, 1997]:

- Scenarios are building on mathematical models (strict quantitative approach) whereby different estimations of the future are calculated down. Simulation models such as “The Limits to Growth” by the Club of Rome are famous representatives of this method.
- Scenarios applied for future studies which are mainly using qualitative approaches for scenario building.
- Development and deployment of a variety of techniques to structure interrelations and interdependencies, as well as to make decisions and choices.

These different scenario approaches have in common a profound analysis of the As-Is situation for identifying causes and effects which serves as initial step. Then key factors are derived from a weight list of impact factors for scenario building [Geschka and Hammer, 1997].

Besides, there are still many ways to classify scenario methods (e.g. [May, 1996], [Glenn et al., 1999], [Van der Duin et al., 2001]) and diverse types of scenarios (e.g. [Van Notten et al., 2003], [Bradfield et al., 2005]). A very detailed classification of scenario projects was made by Gausemeier [Gausemeier et al., 1995].

From this follows that scenario building is an inherently flexible approach in terms of design and construction. It is, therefore, applied in and adopted to many different contexts in both commercial and government organizations [Sharpe and Van der Heijden, 2007]. Kahn and Weiner [Kahn and Weiner, 1967] explain that scenarios describe hypothetical possible (future) events which might occur within an environment. Tool support for scenario building and analysis is not easy realizable as there is no quantitative, logical process behind scenario creation.

However, in the context of the OCOPOMO project tool support is needed in collaborative building scenarios via on-line means and in analyzing the resulting scenarios.

3.3.1. Description of available alternatives for scenario building and analysis

In OCOPOMO, we aim at collaboratively building scenarios by involving different stakeholder groups via on-line means (i.e. scenarios building) and then analyzing the resulting scenarios to derive evidence-based informal rules from narrative descriptions (i.e. scenarios analysis). As scenarios building and scenarios analysis have to fulfil different purposes, each of them needs specific tool support. In the following we will, therefore, distinguish between tools available for scenario building and those available for scenario analysis.

3.3.1.1. ICT support for scenario building

In simple terms it is possible to say that scenarios are just narrative descriptions. Hence, tools to support scenarios building must predominantly facilitate narrative text production. A lot of tools are available to support text production, such as MS Word and LaTeX. Furthermore, scenario building in the context of OCOPOMO project refers also to a collaborative writing process. For scenario building often ICT tools are deployed that are not particularly developed for this purpose. The process of scenario-building can be supported by using collaboration software (Group Support Systems – GSS / Electronic Meeting Systems – EMS / Group Decision Support Systems – GDSS / Collaborative Writing Tools - CWT).

GSS/EMS/GDSS[84] combine computing, communication and decision support technologies to facilitative collaborative work thereby helping to deal with complex, unstructured problems and actors having incompatible interests, diverging areas of knowledge and multiple backgrounds [Van den Herik and de Vreede, 2000]. During meetings in which groups share, structure and evaluate ideas GSS provide support for participants by giving them the opportunity to enter their ideas, reactions or votes to the system that shares these information with the remaining participants. GSS aim at making group meetings and group decision-making more effective [Van den Herik and de Vreede, 2000]. The eGovRTD2020 project used group support software (Ventana GroupSystems) to support the process in

[84] The term GSS is a synonym for EMS and in principle also for GDSS.

the regional scenario-building workshop in Delft, as well as in the validation workshop in Bled [Jansen et al., 2006].

Ventana's GroupSystems software is the recognized leader in meeting support. Winner of Groupware '93 Best of Show Award, and named PC Magazine's Editors Choice, GroupSystems provides the capabilities to capture information and develop consensus as well as make better, faster decisions. Participants interact whenever and wherever they need to work together - in meetings, between offices or around the world. Today, hundreds of organizations worldwide maintain a competitive edge using Ventana's award-winning GroupSystems.[85] GroupSystems was the first GSS that offers such standard functionalities as brainstorming, categorizing, discussing, voting, agenda setting and executing, and recording. The problem with GroupSystems is that it is a LAN-based commercial product, i.e. it is limited to local meetings and for this reason not applicable for open participatory scenario building as intended in OCOPOMO.

Some prototypical web-based GSS, which arose from GroupSystems, are ThinkTank[86], smartSpeed Connect[87] or teambits:workshop[88]. These products differ from GSS in terms of their direction and range of functionality. For instance ThinkTank allows the well-known functionalities from GSS for professional moderated workshops and enlarge it. Teambits provides digital moderation tools for both local meetings and meetings via the Internet. On the other hand, smartSpeed is an integrated set of meeting tools for supporting everyday on-line meetings and workshops, as well as asynchronous working in virtual teams. Focus of developments is on usability even if specific functionalities cease to exist. Besides, a general problem of GSS is that the number of participants is often limited, i.e. that it is not useful to apply these tools for mass and open collaboration as it is intended in OCOPOMO.

GSS are developed to facilitate a group of experts to brainstorm and/or collaborate on a problem. Only a few GSS such as next.moderator[89] and teambits:unite[90] aim at supporting very big groups at big events. Focus of these systems is on best possible networking many people locally and achieving common results thereby automatically protocol the results and make the results quickly available. Because of their ability to deal with large groups of people (i.e. thousands of people) and the fact that the systems are web-enabled, the tools next.moderator and teambits:unite are worth mentioning and considering. However, both systems are proprietary tools. For instance next.moderator is offered as a complete service, which includes the provision and installation of hard- and software as well as technical support and technical moderation during the event. Customers are facilitated with the development of workshop designs and if wanted also with care about moderation of events. So, the application of GSS still focuses on face-to-face meetings (events) and less on mass cooperation and collaboration through the Internet. This is why we decided to not further consider GSS for evaluation and selection for scenario building and to not incorporate them into ICT toolbox.

CWT facilitate the editing and reviewing of a text document by multiple individuals either in real-time or asynchronously. On-line web-based collaborative writing tools such as *Zoho Writer*[91], *Writeboard*[92], *Google Docs & Spreadsheets*[93], *NearTime*[94], *Socialtext*[95], *Quick Doc Review*[96], *EditGrid*[97],

[85] http://www.waria.com/databases/gwvendors.htm

[86] http://www.groupsystems.com

[87] http://www.smartspeed.com

[88] http://www.teambits.de/produkte/teambitsworkshop.html

[89] http://www.nextpractice.de/services/nextmoderator/

[90] http://www.teambits.de/produkte/teambitsunite.html

[91] Zoho Writer is a collaborative editor to create documents and share them publicly or privately. The interface is very intuitive. Any existing document can be imported to work on it. Zoho Writer also enables to export text in several formats. URL: http://writer.zoho.com/

[92] Writeboard is a web-based editor to help writing on-line documents and collaborating with colleagues. Documents can be subscribed via RSS to be notified of changes. URL: http://www.writeboard.com/

SynchroEdit[98], *Please Review*[99], and *Coventi Pages*[100] offer great flexibility. The only requirement is that users must have a well-functioning Internet connection and – depending on the tool – the installation of the respective software. Collaborative writing tools can vary a great deal and can range from the simplicity of wiki system to more advanced systems. Basic features include typical formatting and editing facilities of a standard word processor with the addition of live chat, live mark-up and annotation, co-editing, version tracking and more.

3.3.1.2. ICT support for scenario analysis

Tool support that particularly aims at contributing to scenarios often includes both scenario building and data collection for constructing integrated, long-range scenarios and the respective analysis, e.g. SCrategy Software [Tietje, 2008], The PoleStar System [OECD, 2008] and the Tool for Exploratory Landscape Scenario Analyses (TELSA). Although the objective of these tools is the same, they follow different approaches to best possible fit the specific requirements of their respective problem scopes. The SCrategy Software focuses not only on scenario technique but also supports strategy maps and brainstorming. It supports intergroup development and quantitative analysis. With it, it goes beyond a normal standalone application. SCrategy Software was applied for location promotion, regional and local development, tourism and economic promotion. For scenario analysis it comprises: a) qualitative system analysis (impact matrix, interactive system graph, system grid); b) analysis of feedback loops (from single feedback loops to the most important loops); c) qualitative system simulation; d) consistency analysis; and e) strategy map.

On the other hand, the Scenario Analysis Tool Suite [Dilek, 2009] implemented several scenario analysis techniques, as well as an extended approach of combining methods. Therefore, the tool provides the opportunity to compare different techniques. ART-SCENE [Nam, 2004] includes automatic generation of scenarios from use cases, alternative courses, and guided scenario walkthroughs. From this follows that tools to support scenario building and analysis tend to be as different as the context in which scenarios are applied, such as concerning environmental, social, or economic policy issues. For instance, tools to support scenario analysis for strategic planning base often on a mixture of quantitative calculations enhanced by qualitative aspects. They aim at identifying changes in the environment and the corresponding consequences arising in the long-term

[93] Google Docs & Spreadsheets is a web-based editor to create text documents and spreadsheets. It allows upload of files and makes changes to them on-line available. URL: http://docs.google.com/

[94] NearTime is an on-line asynchronous collaboration platform that supports collaboration. It integrates blogging, wikis, calendaring, email, file sharing, RSS output, tags and more. URL: http://www.near-time.net/

[95] Socialtext is a collaboration platform for working on the same page. Users can edit within a simple WYSIWYG interface thereby previewing while editing and commenting on any page. URL: http://www.socialtext.com/

[96] Quick Doc Review provides an instant private space for gathering comments on any Word and HTML document. It allows commenting on each paragraph, directly within the document. URL: http://www.quicktopic.com/

[97] EditGrid is a service for editing, storing and accessing spreadsheets from any computer with a browser. It allows the import from and export to common formats. URL: http://www.editgrid.com/

[98] SynchroEdit is a browser-based simultaneous collaborative editor with WYSIWYG interface. It supports a simple, text-only editor and clearly depicts user's changes in a specific colour. URL: http://www.synchroedit.com/

[99] Please Review is browser-based collaborative software for reviewing and authoring. URL: http://www.pleasereview.com/

[100] Coventi Pages is an on-line tool that enables users to share, discuss and revise documents. URL: http://www.coventi.com

future. In politics, scenario analysis involves modelling of possible alternative paths of a social, economic, technological or political environment thereby focusing more on qualitative arguable future aspects [Brandao, 2006]. The customization of existing tools for scenario analysis to a specific problem scope using a specific scenario building and analysis approach (e.g. SCrategy applies cross impact analysis, which is not included in the OCOPOMO scenario analysis approach) is the reason why these tools are not applicable for OCOPOMO. Hence, they are not further considered.

In the context of the OCOPMO project tools for scenario analysis shall help identifying latent rules hidden in the narrative descriptions of the scenarios, as well as actors, issues, arguments, tendencies etc. thereby ensuring that the results are traceable. According to [Bicking and Wimmer, 2010], Qualitative Data Analysis (QDA) can be a powerful and inherently flexible approach to tackle different purposes (e.g. future research, system analysis, ethnography, gender research, etc.) across different scientific disciplines (Information Systems, Sociology, Business Science, Psychology, etc.). QDA approaches have proven their value-add in manifold cases.

Since about one decade, CAQDAS (Computer-Assisted Qualitative Data Analysis Software) is developed for supporting the different QDA methods applied in social sciences. These tools might be applicable to support the OCOPOMO scenario analysis approach as socio-scientific approaches aim at investigating phenomena of social interaction within societies. Desk research was carried out to select a small number of CAQDAS for the evaluation. Therefore, a number of existing studies, which compare different CAQDAS, were investigated (cf. [Kuckartz, 2007], [Mayring, 2007], p. 100-108, [Creswell and Maietta, 2002], [Alexa and Zuell, 1999], [Barry, 1998], [Weitzman and Miles, 1995]). Kuckratz ([Kuckartz, 2007], p. 251-257) provides an overview of eight existing CAQDAS (AQUAD[101], ATLAS.ti 5.2[102], HYPERRESEARCH[103], Kwalitan[104], MAXQDA 2007[105], N6[106], NVivo[107], QDA Miner[108], Qualrus[109], and the Ethnograph[110]) and explains general quality criteria such as kind of analysis to be conducted and number of test persons. Mayring [Mayring, 2007]

[101] AQUAD is a software tool that supports content analysis including coding features, memos and word analysis. URL: http://www.aquad.de/eng/index.html

[102] ATLAS.ti helps annotating textual, visual and audio data. It facilitates the categorization process of these types of data and enables the organization of the evolving categories in a (causal) network. URL: http://www.atlasti.com/

[103] HyperRESEARCH™ enables to code and retrieve, build theories, and conduct analyses of data. Works with text, graphics, audio, and video sources through an easy-to-use and flexible interface that facilitates any qualitative analysis technique. URL: http://www.researchware.com/products/hyperresearch.html

[104] Kwalitan supports efficient storage of the data and offers several features to analyse the qualitative material, like coding, retrieving, categorisation of codes, overviews of codes or words in the text, keywords in context and writing memos.URL: http://www.kwalitan.nl/engels/index.html

[105] MAXQDA supports all individuals performing qualitative data analysis and helps to systematically evaluate and interpret texts. It is also a powerful tool for developing theories and testing the theoretical conclusions of the analysis. URL: http://www.maxqda.com/

[106] N6 is the newest version of NUD*IST. It is designed to both code textual data and to efficiently search and navigate research material. URL: http://www.qsrinternational.com/products.aspx

[107] NVivo, a derivative, but not necessarily a replacement for NUD*IST, helps to annotate and organize qualitative data. While it has less coding capabilities than N6, its organizing functions are more elaborate, allowing to link data in a variety of ways. URL: http://www.qsrinternational.com/products.aspx

[108] QDA Miner is an easy-to-use mixed-model qualitative data analysis software package for coding, annotating, retrieving and analyzing small and large collections of documents and images. URL: http://www.provalisresearch.com/QDAMiner/QDAMinerDesc.html

[109] Qualrus is an innovative qualitative data analysis tool for managing unstructured data. It allows for a number of coding strategies, has sophisticated search possibilities, and can handle a variety of data types, and code audio and video data effectively. URL: http://www.qualrus.com/

[110] Ethnograph supports hierarchical coding, text annotations, and advanced data search strategies. URL: http://www.qualisresearch.com/

focuses mainly on ATLAS.ti. Although ATLAS.ti is constructed to meet the specific needs of the theoretical coding according to Glaser and Strauss, Mayring ([Mayring, 2007], p. 100-108) outlines its applicability for the qualitative content analysis (i.e. ATLAS.ti meets the requirements of both the Grounded Theory and the Qualitative Content Analysis). According to Mayring [Mayring, 2007] the most important advantage of ATLAS.ti is its window technique, which displays the protocol and the summary form at the same time and allows the user to work on them simultaneously.

Creswell and Maietta ([Creswell and Maietta, 2002], p. 164 et.seq.) compare seven existing CAQDAS systems (ATLAS.ti, HYPERRESEARCH 2.5, Classic N4, N5, NVivo, the Ethnograph 5, and winMAX) based on the following eight criteria: 1) ease of integration (i.e. logic and layout of the CAQDAS), 2) kind of data the CAQDAS is able to analyse (e.g. text, audio and video), 3) opportunities to read and review data, 4) memo writing, 5) categorisation, 6) analysis inventory and assessment (e.g. search functions), 7) possibility to integrate the analysis of quantitative data, and 8) merging projects (i.e. support of team work). These criteria focus mainly on software engineering aspects while those regarding the research method take a secondary role. Literature review of these comparative studies has shown that experts in the field agree that there is not yet the best candidate found. However, to get from stakeholder-generated scenarios to the models, we will require software similar to ATLAS.ti. As we are committed to using and producing open-source software, ATLAS.ti is not available. However, there are several open source software products available for computer-assisted qualitative data analysis, such as Coding Analysis Toolkit (CAT)111, Digital Replay System (DRS)112 and RQDA113, which are evaluated in sub-section 3.3.3.2.

An alternative is ontology editing software such as Protégé114 except that our concern is largely with social processes and these are not compatible with the ontology approach.

As we aim to raise awareness about the less well-known options, we refer also to tools available in other fields, which can be helpful for OCOPOMO, too. So, argument visualization tools were considered to analyze and evaluate arguments based on audio and video material, as well as written text (such as on-line forums). The software tools shall facilitate structuring and visualization of arguments in various illustration formats, such as graphs or tables. Today, several argument visualization tools exist [Kirschner et al., 2003], for instance ArguMed [Verheij, 2003], Araucaria [Reed and Rowe, 2004], ATHENA [Bertil and Magnusson, 2002], Convince Me [Schank and Ranney, 1995], Compendium [Selvin et al., 2001], Belvedere [Suthers et al., 1995], ProSupport [Prakken and Vreeswijk, 2002], and Reason!Able [Van Gelder, 2002]. These tools produce diagrams using boxes and arrows to link the boxes and to indicate their direction. The boxes represent premises and conclusions, which are formulated as statements.

In the context of OCOPOMO, scenarios are collaboratively developed, i.e. stakeholders are able to discuss on the scenarios, e.g. in discussion forums. This can be seen as a discourse. Some methodological approaches to Discourse Analysis are relatively close to what is needed in OCOPOMO. Within linguistics, analysts ask how written, oral and visual texts are used in specific contexts to make meanings. In particular the socio-political approaches are close to what is intended by OCOPOMO scenario analysis as these approaches focus on the production of knowledge. Political discourse analysis methodological approaches are worth consideration as this analysis focus on the

[111] CAT is a service of the Qualitative Data Analysis Program. In 2008 CAT won the "Best Research Software" award from the organized section on Information Technology & Politics in the American Political Science Association. URL: http://cat.ucsur.pitt.edu/

[112] DRS enables the synchronisation, replay, and analysis of audio and video recordings. URL: http://web.mac.com/andy.crabtree/NCeSS_Digital_Records_Node/DReSS.html

[113] RDQA is an R package for Qualitative Data Analysis. URL: http://rqda.r-forge.r-project.org/

[114] Protégé is a free, open source ontology editor and knowledge-base framework. URL: http://protege.stanford.edu/

analysis of informal exchange of reasoned views as to which of several alternative courses of action should be taken to solve a societal problem. As this can be important for policy modelling and the derivation of formal rules, this category of tools is mentioned. However, the focus of discourse analysis tools is too much on linguistic issues thereby neglecting that our concern is largely with social processes and these are not compatible with the linguistic approach.

Argument visualisation and discourse analysis tools can help to structure and analyse textual arguments. These tools can be used to transform arbitrary argument structures into both graphical and/or text-based summaries. These tools can be used for drawing abstract argumentation frameworks, which are simply debate graphs. In general these tools are not applicable for scenario analysis as they are designed for a different purpose based on different methodological and disciplinary context. These tools are too much customized for discourse analysis and, thus, too inflexible for an in-depth scenario analysis that detects the underlying social processes as intended in OCOPOMO. Both discourse analysis and argument visualisation tools are still in an early stage with much room for improvement and are not easily adoptable and applicable to meet the scenario analysis requirements of OCOPOMO. For the visualisation of some scenario analysis results it might be interesting to further consider these tools but in the end none of these categories of tools meet the criteria for scenario analysis to the same degree as CAQDAS systems do. This is why no evaluation is presented for argument visualization and discourse analysis tools.

3.3.2. Definition of criteria for selecting tools to incorporate into ICT toolbox

Initial statements of needs and rations that must be used in evaluating the decision for a specific tool or the combination of tools to form correct judgments regarding the tool support of the intended scenario building and analysis are made in D1.1 [Bicking et al., 2010]. The requirements formulated in D1.1 ground the definition of criteria for selecting tools for scenario building and scenario analysis. Since production of D1.1, the vision of open participatory scenario building and policy modelling made a progress and takes more and more shape. The definition of criteria will, thus, build on this progress.

3.3.2.1. Definition of criteria for selecting tools for scenario building

Scenario building aims at producing narratives that provide the fundamental basis for policy model design. This means that criteria for scenario building are strongly related to criteria that are applicable for collaborative writing tools. The key features important for scenario building are [Kolabora, 2007]:

Functionality	**Description**
CWT[115]	Key functionalities of CWT
Text chat **(**T-1, T-4, T-12, I-5)	The presence of a text chat that users can utilize to communicate while editing
Versioning (T-5, I-11)	The capability to track all changes made to the original document and go back to older versions
RSS (T-24, T-30)	Support for RSS feeds, allowing users to get real-time notifications when changes are made
Email updates (T-24, T-34)	The capability to receive email updating users when there are changes to the documents that are being edited

[115] Identifiers in parentheses represent IDs of user requirements as identified in D1.1 [Bicking et al., 2010]

	Public/Private (T-37, I-4)	The possibility to set up private or public collaborative editing sessions
	Web-based (I-7, I-NF-7)	The type of collaborative system should be web-based
	Comments (T-25)	The possibility to add comments to the document
	Expected familiarity (I-NFT-1, I-NFT-8)	The degree to which a user recognizes user interface components and views their interaction as natural; the similarity of the interface to concrete objects the user has interacted with in the past

Table 9 Criteria for selecting tools for scenario building derived from requirements identified in D1.1

The following features are not directly corresponding to user requirements identified in D1.1. They propose new requirements that came up while progressing with the definition of the scenario building process.

Functionality		**Description**	**Reason**
CWT		Key functionalities of CWT	Reasons for new requirements
	File types supported	What are the supported file formats for the files that can be edited collaboratively	For scenario building it is important that the CWT supports the file format to be decided on
	Max editors	The maximum number of editors allowed	As mass-collaboration is wanted, the tool to be used should allow for a certain high number of editors (persons)
	Real-time co-editing	The possibility to collaboratively edit in real-time, i.e. several people shall edit the document at the exact same time, making different changes in real-time with very little latency	It is a crucial quality criteria in terms of usability that the tool allows multiple editors at once to build the scenario(s)
	Price	Open source software (OSS) or proprietary	The final ICT toolbox shall be available for free for everyone to use and improve the system, therefore, it is necessary to select only OSS tools as components of the system including the ICT toolbox

Table 10 Criteria for selecting tools for scenario building that came up while progressing with the definition of the scenario building process

In addition to CWT, several e-participation tools (in particular the CMS category) are available that support either single criteria or a collection of these criteria (cf. section 3.2 on e-participation tools and technologies).

3.3.2.2. Definition of criteria for selecting tools for scenario analysis

Analysis of scenarios aims to identify and extract relevant information and parameters for policy modelling. The following features are important for supporting the scenario analysis:

Functionality	Description
CAQDAS[116]	Key functionalities of CAQDAS
Coding of text (T-39, FR01_PM, FR03_PM, FR04_PM, FR05_PM)	To assign or better to link keywords to text passages thereby ensuring that a keyword clearly describes the meaning of a text passage We distinguish between the following features [Koenig, 2010]: • **In vivo coding**, i.e. assign the text that is to be coded to a code, whose label is the text itself. While this is a very efficient method for coding, there might be theoretical consideration to use this option cautiously. • **Contextual coding,** i.e. after searching your data for certain text and/or codes, you might jump to your finds and code them in context. **Automatic coding** allows the user to perform text and/or code searches and assign a code to the search results.
Recovery of the coded position (T-39)	Recovery of the coded position within the text material to avoid de-contextualization and ensure traceability of results
Cluster codes (T-39, FR01_PM, FR03_PM, FR04_PM, FR05_PM)	Advance codes by combining those that have the same and/or similar meanings to superior codes[117] Most CAQDAS systems allow annotating codes in a variety of ways.
Code relations (T-39, FR01_PM, FR03_PM, FR04_PM, FR05_PM)	Code relations between codes
Depict the net of issues and relations (NFR01-PM, related to I-39)	Depict the net of issues and relations. All codes are usually stored in a codebook. In some CAQDAS the codes can be structured in a hierarchy and/or a network. Some CAQDAS also allow colouring codes to organize them [Koenig, 2010].
Flexible collecting several additional data (FR02_PM, FR03_PM, FR04_PM, FR05_PM)	Flexible collecting several additional data related to the code such as personal background, government level. This information is necessary to avoid removing the issues from their original context.

Table 11 Criteria for selecting tools for scenario analysis derived from requirements identified in D1.1

The following features are not directly corresponding to user requirements identified in D1.1. They propose new requirements that came up while progressing with the definition of the scenario building process.

[116] Identifiers in parentheses represent IDs of user requirements, identified in D1.1 [Bicking et al., 2010]

[117] In social sciences the term “categories” is used to classify broad groups of similar concepts that are used to generate a theory. Concepts are collections of codes of similar content that allows the data to be grouped. So, the term “concept” classifies superior codes or better clusters of codes. The OCOPOMO nomenclature uses the term “issues” to classify a cluster of codes concerning similar contextual aspects

Functionality		**Description**	**Reasoning**
CAQDAS		Key functionalities of CAQDAS	Reasons for new requirements
	Information structuring, querying and presentation	Identifying and structuring the information extracted from the unstructured scenarios as needed for the CCD and policy modelling, e.g. the social network, the social process, etc.	The tools applied for scenario analysis shall deliver results valuable for the development of the CCD and simulation model
	Memos	Record comments and questions to text passages, codes and/or categories thereby ensuring that a linkage is built between comment and questions and the corresponding text passages, codes or categories	Memos are needed to support the communication inside the team of analysts for facilitating better and common understanding of analysis results e.g. if results are not self-explanatory
	Price	Open source software or proprietary	The final ICT toolbox shall be available for free for everyone to use and improve the system, therefore, it is necessary to select only OSS tools as components of the system including the ICT toolbox

Table 12 Criteria for selecting tools for scenario analysis that came up while progressing with the definition of the scenario analysing process

3.3.3. Evaluation of tools

In this subsection two overviews of information about tools are provided – one for scenario building and the other one for scenario analysis. The overviews are designed to help make informed choices between tools, to plan for their effective use and to manipulate the tools in creative ways to meet methodological and practical needs of OCOPOMO scenario building and analysis.

The reviews provide up-to-date comparative information about selected tools for both scenario building and analysis. The reviews include both commercially available and free/open source products. We aim at introducing the leading tools which are well established in the respective fields. Reviews of ICT support for scenario building and analysis do not provide an exhaustive account of all the features and functions provided by the tools but are designed to highlight the key features important for OCOPOMO scenario building and analysis. The comment section at the end details certain aspects we consider as worth knowing.

3.3.3.1. ICT support for scenario building

This sub-section focuses on the evaluation of collaborative writing tools for scenario building. The next table provides an overview of several collaborative writing tools according to the criteria formulated in subsection 3.3.2.1. The collaborative writing tools presented merely a selection of the tools available but do not claim for completeness.

Tools/ Criteria	**Zoho Writer**	**Write-board**	**Google Docs & Spread-sheets**	**NearTime**	**Socialtext**
File types supported	Text, Images	Importing external files is not possible	Text, Spread-sheets, Images	Text, Images	Text, Images
Text chat	No	No	Yes	No	No
Versioning	Yes	Yes	Yes	Yes	Yes
RSS	No	No	Yes	Yes	Yes
Email updates	Yes	Yes	No	Yes	Yes
Public/Private	Public/Private	Public/Private	Public/Private	Private	Private
Max Editors	N/A	Unlimited	50	Unlimited	Unlimited
Real-time co-editing	No	No	Yes	No	No
Web-based	Yes	Yes	Yes	Yes	Yes
Comments	No	No	Yes	Yes	Yes
Export/File formats	Yes	No	Yes	Yes	Yes
Price	Free	Free	Free	Proprietary	Free
Familiarity	No	No	Yes	No	No

Tools/ Criteria	**Quick Doc Review**	**EditGrid**	**Synchro Edit**	**Please Review**	**Coventi Pages**
File types supported	Text	Spread-sheets	Text	Text, Images	MS Word
Text chat	No	Yes	Yes	No	No
Versioning	Yes	Yes	No	Yes	Yes
RSS	No	No	No	No	No
Email updates	Yes	No	No	N/A	Yes
Public/Private	Public/Private	Public/Private	Private	Private	Private
Max Editors	N/A	N/A	N/A	N/A	N/A
Real-time co-editing	No	Yes	Yes	No	No
Web-based	Yes	Yes	Yes	Yes	Yes
Comments	Yes	No	No	Yes	Yes
Export/File formats	Yes	Yes	No	No	Yes
Price	Proprietary	Proprietary	Free	Proprietary	Free
Familiarity	No	No	No	No	No

Table 13 Evaluation of collaborative writing tools based on key features [Kolabora, 2007]

All collaborative writing tools evaluated are more or less suitable and worth considering for the implementation of the toolbox. But financial issues and usability criteria should be considered, too. Only five out of ten investigated tools are free and open source. Besides, only the basic editions of two further tools are free and open source. As the toolbox shall be available for free at the end of the project in order to not discriminate poor countries from usage, those tools which are proprietary are rejected from further considerations.

From user perspective, usability including in particular familiarity with and learnability of the tools deployed play a central role for participating in the scenario building process. Hence, using well-known e-participation tools such as on-line forums, wikis or blogs for scenario building is likely to succeed in attracting people to participate. This means also that a system deploying unfamiliar collaborative writing tools is likely to fail in establishing long-lasting interest of ordinary people to participate. Scenario building takes a long time to do, thus, it is wise to deploy tools that require little training and/or explanation time. The effort it takes people to learn working with unfamiliar collaborative writing tools is likely too high in comparison to the benefit stakeholders may perceive through participation. If learning time is the typical measure, tools (i.e. user interfaces) are typically easier to learn when they are familiar. Familiarity may come from using tools people already use and made experiences with. This is why we propose to use a combination of e-participation tools to provide the key features required because many people experienced these tools over the last years in the course of the boom of using web 2.0 technologies in private life and every day business. People are social beings, i.e. sometimes it really does not matter how magnificent the tool is; if they do not know how to use it, they just will not use it. But for a long term sustainable engagement it is necessary that people spend their time with building scenarios and not with learning how to express themselves.

Besides, CMS as mentioned in section 3.2 provide also much functionality that are valuable and worth consideration for collaborative scenario building, in particular the support of workflows and versioning. This tool category is therefore also applicable as it allows the following functionalities: personalised profiles, rating content, on-line meetings and chats, comment content, moderated and non-moderated discussions, visibility of discussions for certain user roles, multiple instances of a forum, possibility to order entries in chronological order and for topics, rating of contributions and contributors, notification via RSS feed and email, different types of questions & answers, graphical presentation of the results. The fact, that all evaluated CMS tools are open source and meet the basic requirements identified in D1.1 and criteria derived and presented in section 3.3.2.1, grounds the recommendation to further consider CMS for the scenario building implementation instead of CWT.

3.3.3.2. ICT support for scenario analysis

This subsection focuses on the evaluation of CAQDAS for scenario analysis. The next table provides an overview of several CAQDAS systems according to the criteria formulated in subsection 3.3.2.2. The CAQDAS presented merely a selection of the most popular and advanced tools available but do not claim for completeness.

Tools/ Criteria	**ATLAS.ti 5.0 RC2**	**HyperRE-SEARCH 2.6**	**Kwalitan 5.0**	**MAXqda 2k3**
In vivo	Yes	No	Yes	Yes
Contextual	Simple	Difficult	Simple	Difficult
Automatic	Very slow	Difficult	Simple text only	Quick and stable
Recovery of code position	Yes	Yes	Yes	Yes
Cluster codes	Yes	Yes	Yes	Yes
Code relations	Yes	No information	No information	Yes
Visualising	Flat, but variable	Flat	Complex tree	Hierarchical

network structure	linked codes		structure	
Information structuring, querying and presentation	Not as intended in OCOPOMO	Not as intended in OCOPOMO	Not as intended in OCOPOMO	Not as intended in OCOPOMO
Memos	Yes	Yes	Yes	Yes
Flexible data collection	Not as intended in OCOPOMO	Not as intended in OCOPOMO	Not as intended in OCOPOMO	Not as intended in OCOPOMO
Price	Proprietary	Proprietary	Proprietary	Proprietary
Operating system	Windows XP / Vista / 7	Windows 2000 / XP / Vista / 7 Mac OS X (PowerPC or Intel)	Windows XP / Vista	Windows XP / Vista / 7
RAM	2 GB	2 GB	8 MB	2 GB
Free disk space	50 MB	41 MB	5 MB	130 MB

Tools/ Criteria	**QSR NVivo 2.0**	**Coding Analysis Toolkit**	**Digital Replay System**	**RQDA**
In vivo	Yes	Yes	Yes	Yes
Contextual	Simple	Simple	Simple	Simple
Automatic	Very slow	Yes	No	No
Recovery of code position	Yes	Yes	Yes	Yes
Cluster codes	Yes	No	Yes	Yes
Code relations	Yes	Yes	Yes	Yes
Visualising network structure	Hierarchical, diverse codes	Not as intended in OCOPOMO	Not as intended in OCOPOMO	Not as intended in OCOPOMO
Information structuring, querying and presentation	Not as intended in OCOPOMO	Not as intended in OCOPOMO	Not as intended in OCOPOMO	Not as intended in OCOPOMO
Memos	Yes	Yes	Yes	Yes
Flexible data collection	Not as intended in OCOPOMO	Not as intended in OCOPOMO	Not as intended in OCOPOMO	Not as intended in OCOPOMO
Price	Proprietary	GPLv3 licence	BSD licence	BSD licence
Operating system	Windows 2000 / XP	Windows XP / Server 2003 / Vista / 7	Windows XP Mac OS X	No information
RAM	128 MB	2 GB	2 GB	2 GB
Free disk space	125 MB	No information	No information	No information

Table 14 Evaluation of computer-assisted qualitative data analysis software based on key features (retrieved from [Koenig, 2010] and [Surrey, 2010])

The individual reviews of CAQDAS are intended to be read in conjunction with the references and information provided in subsection 3.3.1.2, which together provide a broader comparison of CAQDAS functionalities. All CAQDAS systems evaluated are more or less suitable and worth considering for the implementation of the toolbox. But financial issues should be considered, too. As the toolbox shall be available for free all tools which are proprietary are rejected from further considerations. However, the most important requirement is that the scenario analysis produces high-level input for creating conceptual descriptions and the simulation model. As this requirement is not met by any of the CAQDAS and cannot be met by any other tool available so far, it was decided to conceptualise and implement an analysis tool during the project runtime that will meet the exact requirements of OCOPOMO scenario analysis and those requirements to support the integration of stakeholder-generated scenarios and formal models.

The final knowledge is not yet generated of how to design the integration of stakeholder-generated scenarios and formal models. This means that OCOPOMO may have to modify specific features based on requirements not yet known. In this context, we assume that none of the existing tools is flexible enough in the manner OCOPOMO may need. Since a tool under our own control can be more easily and quickly adopted (if requirements are changing over time) than the ones we did not implement by ourselves, we propose conceptualising and implementing a tool for covering the scenario analysis and the transformation process towards the simulation model.

3.4. FORMAL MODELLING TOOLS AND TECHNOLOGIES

Formal modelling in the context of the OCOPOMO project refers to a process of abstraction that turns narrative descriptions of policy measures and their impacts into precise, formal statements that are isomorphic with logical theorems. The models are agent-based so that each software agent represents an individual or an organisational or collective stakeholder as may be deemed appropriate in the specific social context.

As discussed in the DOW [Ocopomo-DoW, 2009], one of key advantages of agent simulation over other paradigms like system dynamics or queuing models [Gilbert and Troitzsch, 2005] is that agents capture relevant aspects of how people think and behave both individually and when working together, while other models reflect doctrinal standard operating procedures, how machines should operate or laws of physics. On the downside, major drawbacks associated with using agent simulation are the complexity of the resulting control system that needs to be debugged and the lack of facilities to adequately represent/trace knowledge contained by each agent and the selection of tactics used by the agents.

In this project as in several projects before it (FP6 projects CAVES[118], EMIL[119], FP5 project FIRMA[120]), the behaviour of the agents will be modelled declaratively. This means it is driven by rules that capture as far as possible relationships described by stakeholders in their own linguistic terms. The virtues of this approach include:

- The models and the behaviour of the agents can then be validated at micro level by seeking evaluations from the stakeholders who know the persons or collectives represented by the agents.
- Numerical outputs from the model can be produced for comparison with analogous, real social data.
- The agent rules can produce text explaining the reasons for actions taken by the agents where such explanations are drawn from the conditions of the rules that produce the actions. The

[118] http://www.cfpm.org/caves

[119] http://emil.istc.cnr.it/

[120] http://cfpm.org/firma/

result is a running narrative about and by the agents in the models and the consequences of their actions. This output amounts to a formally generated scenario.

- The agent-based model structure offers scope for software agents to be replaced in the simulation runs by human users.

While it would be possible to develop and implement such formal models in any general-purpose programming language, it is only sensible to apply existing toolkits particularly suited to the purpose of building (complex) agent-based simulation models.

It is our understanding that declarative modelling is often the most appropriate technique to capture social phenomena [Moss and Edmonds, 2005] whereas many physical or biological processes are best described by numerically-based formalisms. Since the models developed in the OCOPOMO project need to represent both, social and physical processes, it is therefore important that a formal modelling environment should support both declarative and imperative/procedural programming paradigms.

Due to the lack of integrated software for this policy modelling approach, which would be capable to cope with the anticipated complexity, models developed in the context of the OCOPOMO use cases certainly will probably involve more extensive functionality than any single existing tool can provide. As a consequence, the OCOPOMO policy modelling tool will be composed of several frameworks and components, each covering a specific set of functionality.

3.4.1. Description of available alternatives

This section presents the state of the art in (i) agent-based simulation platforms and (ii) rule engines. Both need to be combined in OCOPOMO to allow for declarative, i.e. rule-based policy modelling as set out in Deliverable 1.1 [Bicking et al., 2010]. While an agent-based simulation platform provides the necessary functionality to build and execute agent-based models, a rule engine adds the functionality to define the agents' behaviour in terms of rules. A combination of these tools will therefore provide the integration of declarative features within agent-based social simulation software. There are several options to achieve the combination of rule engine and agent-based model, ranging from using one rule engine per agent to sharing not only the rule engine but the complete rule base amongst all agents within a model. All the options have their advantages and disadvantages regarding e.g. memory requirements, execution speed and conceptual clarity [Caves, 2006].

In the following, we will constrain the discussion to toolkits which are available free and open source as this is the foremost requirement for the integrated ICT toolbox to be developed in OCOPOMO.

3.4.1.1. General agent-based simulation platforms

Over the past decade, a variety of agent-based simulation platforms have emerged. While some of them are built for a particular domain, ranging from education to battlefield simulation [Berryman, 2008], a number of them are aimed at general-purpose modelling. The following gives an overview of the main general-purpose agent-based simulation platforms that are available for free and (more or less) open source[121]. For a comprehensive review of the currently existing agent-based simulation

[121] An example of a commercial simulation platform can be AnyLogic (http://www.xjtek.com/) currently in version 6.5.1 combining agent based modeling with process based and system dynamics modeling approaches.

toolkits see [Nikolai and Madey, 2009]; other reviews focus on a smaller selection of toolkits (e.g. [Railsback et al., 2006], [Tobias and Hofmann, 2004], [Gilbert and Bankes, 2002]).

Swarm

Swarm was originally developed at the Santa Fe Institute [Minar et al., 1996] and is now developed by the Swarm Development Group. It is the “ancestor” of many of the current ABM (Agent-Based Modelling) platforms. The basic architecture of Swarm is the simulation of collections of concurrently interacting agents (“swarms”), and this paradigm is extended into the implementation, including agent inspector actions as part of the set of agents. Swarm is a stable and widely used platform, and seems particularly suited to hierarchical models. As such, it supports good mechanisms for structure formation, through the use of multi-level feedback between agents, groups of agents, and the environment, which are all treated as agents.

The Objective C Swarm requires learning Objective C, which can be a difficult language for inexperienced programmers. The Java version of Swarm feels cumbersome, and is worse than the Objective C Swarm in terms of documentation and code examples [Berryman, 2008].

Repast

Repast, the Recursive Porous Agent Simulation Toolkit [North et al., 2006], is a widely used, free, and open source agent-based modelling and simulation toolkit. While the main version is Java-based (Repast-J), two other versions have been released: Repast for the Microsoft .NET framework and Repast for Python scripting.

Repast provides modellers with a framework for agent-based simulations along with a large variety of libraries for the representation of environments (grid, network, topography), mathematical operations (e.g. statistical analysis, random number generation), visualisation (diagrams, animations etc.) and some basic AI functionality like neural networks and genetic algorithms. Repast-J can easily be extended with any functionality available for Java-based applications. A recent version (Repast Simphony) is intended to provide an alternative in form of graphical control flow design in conjunction with an alternative modelling language (Groovy), but more complex or “beyond standard” models still require the use of general-purpose programming languages.

Mason

MASON (“Multi-agent Simulator Of Neighbourhoods / Networks") is a general purpose ABM library, which is geared towards speed and portability. It is implemented in Java. While MASON provides many of the same features as Repast, its core has been kept deliberately small, making use of pre-existing libraries instead (e.g. JFreeChart to produce charts and graphs). According to the developers, “MASON carefully delineates between model and visualization, allowing models to be dynamically detached from or attached to visualizers, and to change platforms mid-run” [Luke et al., 2005].

Primarily, the advantage of MASON is in speed, however it is faster than Repast by only a small amount, and for some models is slightly slower than Repast [Railsback et al., 2006]. A strong point in favour is that MASON guarantees the replicability of model runs, i.e. it can produce results that are identical across platforms.

Ascape

Ascape [Parker, 2001] is another general-purpose toolkit for agent-based simulation. It was originally developed at the Brookings Institution as software for the seminal Sugarscape model [Epstein and Axtell, 1996]. As the majority of ABM toolkits, Ascape is implemented in Java. It provides functionality similar to Swarm (“scapes” as collections of agents instead of “swarms”) and Repast.

Ascape claims to be “designed to be flexible and powerful, but also approachable, easy to use and expressive”[122].

NetLogo

NetLogo is a “multi-agent programmable modelling environment” [Wilensky, 1999], aimed to support users in rapidly creating models and running experiments with them. It is easy to handle and requires the least programming experience of all reviewed toolkits, but shows certain restrictions, e.g. a grid-based environment. Agent behaviour is described through a functional language with no further means for structuring code, which results in the disadvantage that agent descriptions cannot be separated from technical details of algorithms and user interfaces which are of no interest to stakeholders. The documentation and number of example models for NetLogo are both excellent, as is the user community, which provides a lot of support to new users.

NetLogo itself is implemented in Java and provides APIs for controlling it from external code and extending the language with new commands and reporters; this makes extension possible albeit somewhat difficult. For the future, a release as open source is planned[123].

Table 15 gives an overview of the technical details of the presented agent-based simulation platforms.

Platform	Swarm	Repast	Mason	Ascape	NetLogo
Latest version	2.2	3.1	14	5.6.0	4.1.1
URL	http://www.swarm.org/	http://repast.sourceforge.net/	http://cs.gmu.edu/~eclab/projects/mason/	http://ascape.sourceforge.net/index.html#Introduction	http://ccl.northwestern.edu/netlogo/
License	GPL (GNU General Public License)	BSD	AFL 3.0 (Academic Free License)	BSD	Personal licence (free software, code modification granted for educational/research purposes, source code not yet available)
Category	Library	Library	Library	Library	Simulation Environment
Programming language	Objective C (Java)	Java (Python, C#)	Java	Java	Logo dialect

Table 15 Basic characteristics of the selected ABM platforms

3.4.1.2. Rule engines (rule-based systems)

The following discusses a representative sample of currently available rule engines. Jess is by far the most stable, comprehensive and most widely used rule engine. A number of other, open-source rule

[122] http://ascape.sourceforge.net/index.html#Introduction

[123] http://ccl.northwestern.edu/netlogo/faq.html

engine projects have emerged[124], partly implementing the standard Java Rule Engine API [jsr-94, 2004]. Most of these projects, like Jess, make use of the RETE algorithm [Forgy, 1982] to compile the rule bases in order to speed up performance.

Jess

Jess, the Java Expert System Shell, is a rule engine and scripting environment developed at Sandia National Laboratories by Ernest Friedman-Hill. Although not open source, it is available free of cost for academic purposes, including the source code. Since it is written entirely in Java and allows for calling Java methods from rules, it integrates well with any Java software.

Jess consists of a rule interpreter which can apply both forward and backward chaining, using an improved version of the fast but memory-intensive RETE algorithm to match facts from the fact base to rules in the rule base. Declaring facts and rules is done via a script language with a LISP-like syntax. This language supports not only the manipulation of symbolic facts but also method calls on arbitrary Java objects, thus facilitating the combination of declarative modelling and imperative modelling. Jess has a wide and active user community, with good documentation and support by the developer.

JRuleEngine

JRuleEngine is a forward-chaining rule engine, i.e. the engine implements an execution cycle that allows the action of one rule to cause the condition of other rules to become met. In this way, a cascade of rules may become activated and each rule action executed. Forward-chaining rule engines are suitable for problems that require drawing higher-level conclusions from simple input facts.

JRuleEngine is based on the Java Rule Engine API [jsr-94, 2004], i.e. rules can be retrieved from an XML file or can be stored via JRuleEngine APIs, so rules could be stored in any kind of external storage, like a database. The distribution consists of a library that can be embedded into any Java application.

When evaluating JRuleEngine as a candidate for rule engines in OCOPOMO we found it to be unsuitable due to its internal implementation, which allows only one instance per fact type at any one moment in time.

JEOPS

JEOPS [Figueira and Ramalho, 2000], the Java Embedded Object Production System, is a declarative rule engine, which extends the Java programming language with a mechanism for embedding first-order, forward-chaining production rules. It does implement the RETE algorithm and, thus, is optimized for application in expert systems. With restrictions, it can be (and also has been) used for simulation purposes.

JEOPS is no longer supported by its developers and does not have an active user community.

Table 16 gives an overview of the technical details of the discussed rule engines:

Rule engine	Jess	JRuleEngine	JEOPS
Latest version	7.1p2	1.3	2.1.2
URL	http://www.jessrules.com	http://jruleengine.sourceforge.net/	http://www.di.ufpe.br/~jeops/

[124] http://java-source.net/open-source/rule-engines

License	Available free for academic use; redistribution of the source code under any open source license is prohibited	GPL	Unspecified open source
Category	Rule engine	Rule engine	Rule engine
Programming language	Jess script language, Java	Java	Java, own rule specification language

Table 16 Basic characteristics of the selected rule engines

3.4.2. Definition of criteria for selecting tools to incorporate into ICT toolbox

The criteria for selecting tools to incorporate into the ICT toolbox can be divided into more general criteria and criteria specific to the OCOPOMO project. Important criteria in general are the following ([Najlis et al., 2001], [Railsback et al., 2006]):

Open source	The ability to obtain the source code. It enables users to extend the facilities provided by the software, or to add in other platforms to provide missing facilities. This can also be of importance where there are bugs in the platform, or if the documentation is poor.
Flexibility	This is the ability to write custom agents and agent behaviours (no limitation to a set of predefined possibilities to select from).
Speed	Speed of execution is important, particularly under statistical replication and also when a variety of scenarios or parameters need to be explored.
Support	Support is important in order to fully use the platform or to even start to use the software. The support provided by user communities can also be of high importance and this is considered along with documentation.
Facilities	The facilities the software provides e.g. for drawing graphs, recording simulation data to file for further analysis, etc.
Scalability	Possibility to perform mass simulations (can deal witch a big number of agents, big number of rules/facts, etc.
Extendability	Important for possible extension and/or customisation of the tool (e.g. adding code written in a general programming language).

Table 17 Criteria for selecting tools for agent-based formal modelling

3.4.3. Evaluation of tools

General agent-based simulation platforms

There is a range of agent-based simulation platforms, some of which have small user communities and depend on a single developer for maintenance and support. We have restricted our evaluation to widely used platforms with substantial development teams.

Repast is currently available in two versions: Repast 3.1 and Repast Simphony. Both are Java libraries that provide user interfaces for agent-based simulation models. Whilst it is straightforward to implement agents in Repast, there are no constraints on how the agents are implemented as long as they are implemented in Java. Repast provides a framework to activate the agents as well as facilities for producing graphs of various kinds to report and/or summarise numerical measures or numerically defined outputs from agent activity. Repast Simphony, in addition, provides a graphical interface for implementing agents.

MASON has more recently been developed as an alternative to Repast with the intention of making the platform more scalable and faster in its use of mathematical operations. Like Repast, it is implemented as a set of Java libraries. It has a much smaller user base. As in Repast, the implementation of agents is independent of and unconstrained by the platform.

NETLOGO has a much more elaborated graphical interface for designing and implementing agents and for controlling behavioural parameters with sliders and menus that are not bundled with either Repast or MASON. However, it constrains the implementation of agents to render the design consistent with the top-level interface with its sliders and menus and these constraints render it unsuitable for declaratively represented agents. It is therefore not suitable for use in OCOPOMO.

ANYLOGIC is a commercial, closed-source modelling platform and is, for that reason alone, excluded for consideration in OCOPOMO. Like Repast and MASON, the implementation of agents is not highly constrained and it also has specific support for system dynamic modelling. System dynamics was not included in the OCOPOMO software requirements analysis so restricting ourselves of open-source software, thereby to reject the use of AnyLogic, has no cost in terms of required functionality.

The only attractive and viable candidates for the simulation platform to be used in OCOPOMO are the two versions of Repast and MASON. We have selected Repast because it has been in development for a much longer period of time than MASON and it has a much wider user base with a wider range of applications. Consequently, it seems more likely that instabilities and bugs have been found and removed. Certainly, in using Repast 3.1 for more than five years, no programming or design problems have been encountered. Also, numerical calculations are not dominant in the OCOPOMO models since we have chosen to implement agents around a rule-based, declarative design. As between Repast 3.1 and Simphony, we have decided to stay with Repast 3.1 because the lack of additional graphical functionality and the additional layers of software required to support the graphical interfaces. Nonetheless, declarative agent software to be used should be implemented in such way that it will run on each of these platforms linked to them by an abstract model class.

Rule engines

Existing rule-based systems are mainly optimised for expert systems where the fact base changes infrequently or not at all. To take advantage of this property, this type of software frequently uses algorithms that compile the rule base (in most cases: a variant of the RETE algorithm [Forgy, 1982]). If a fact is added, modified or deleted, then the rule base is recompiled. In a simulation environment where the fact bases represent the working memory of agents and are changing frequently, recompiling becomes more or less continuous and, therefore, time-consuming as well as highly memory-intensive. Having experimented with several RETE-based rule engines like JESS and JEOPS, we have confirmed that the class of RETE-based declarative and rule-based systems are inappropriate for social simulation in general. We have not been able to find any currently supported declarative systems that are not based on the RETE algorithm. Two declarative, rule-based agent modelling systems, SDML [Moss et al., 1998] and DESIRE [Brazier et al., 1997], that are not based on the RETE algorithm are no longer supported and the programming code is impenetrable. Several OCOPOMO partners, Moss (SMA) and Meyer (MMU) are well experienced in declarative modelling and Moss in particular produced the rule-based software that evolved into SDML.

On the basis of this experience, a decision was taken not to use any existing rule engine but to implement a new one better suited for social simulations. Therefore, we are now developing a **d**eclarative, **r**ule-based, **a**gent **m**odelling **s**ystem (DRAMS) that has the efficiency properties of SDML. A number of OCOPOMO project partners (SMA, UKL, MMU) are involved in the development of this software within the scope of WP5 [Moss et al., 2010].

Each agent has a rule base and a fact base. The rules govern agent behaviour and interaction. Because agents are not universally informed about every aspect of the system state, they have different perceptions, which, via rules, lead them to add different facts or logic-like clauses to their databases. As a result and even within the same class of agents, different fact base contents evolve so that the agents behave differently and generally uniquely. This is a very flexible and, if properly implemented, scalable approach to agent-based social simulation in general and policy modelling in particular. To achieve the required speed of execution, rule bases will be compiled on dependency digraphs where each link indicates that conditions on the left hand side (LHS) of the rule represented by the to-node are satisfied if the right hand side (RHS) of the rule represented by the out-node has been executed.

The development of rule bases is facilitated by being able to run a model, stop it and then explore the fact base of any agent. The proposed system includes facilities for supporting the modelling and simulation process (interactive generation and management of rule and fact bases, calculating and visualization of dependency graphs). By writing conditions line by line and then determining whether they are satisfied, the LHS of a rule can be developed and tested incrementally within a known system state.

3.5. TOOLS PRESELECTON AND NEW REQUIREMENTS IDENTIFICATION

3.5.1. Additional requirements

Different tools have been evaluated in order to investigate degree of match between the tools and criteria we consider relevant for OCOPOMO. Most criteria were based on (or inspired by) user requirements presented in [Bicking et al., 2010]. In addition to these criteria, several criteria have been used which do not reflect defined user requirements. They are based on our current understanding of processes standing behind policy modelling. In order to reflect the evolution of this understanding, these additional criteria must be transformed into new requirements. As a result, the following new requirements have been defined:

Requirement ID: SOTA-1 **Requirement Type:** Functional **Priority:** Should-have
Name: Workflow engine
Description: Workflow engine to manage sequences of activities e.g. publication and review workflows for documents, forum entries, scenarios etc.
Measurement indicators: Available functionality.

Requirement ID: SOTA-2 **Requirement Type:** Functional **Priority:** Should-have
Name: Content/ WYSIWYG
Description: Texts should be editable by users with the help of an editor enabling ‘what you see is what you get’ editing manner.
Measurement indicators: Available functionality.

Requirement ID: SOTA-3 **Requirement Type:** Functional **Priority:** Should-have
Name: File types supported
Description: Several file types support enables to manipulate and/or retrieve information from document sources using different formats. It broadens possible sources to be utilised as information resources.
Measurement indicators: Possibility to read/write in file formats required by pilot applications.

Requirement ID: SOTA-4 **Requirement Type:** Functional **Priority:** Should-have
Name: Several document editors
Description: Texts should be allowed to be edited by a certain number of human editors. The maximum number of editors should be set in a way enabling open collaboration over texts (an exact number limit to be decided after gaining deeper understanding of OCOPOMO processes).
Measurement indicators: Available functionality of a number of editors.

Requirement ID: SOTA-5 **Requirement Type:** Functional **Priority:** Nice-to-have
Name: Real-time co-editing
Description: Texts should be editable by several users collaboratively in the same time, making different changes in real-time.
Measurement indicators: Available functionality.

Requirement ID: SOTA-6 **Requirement Type:** Functional **Priority:** Should-have
Name: Information structuring
Description: Identifying and structuring the information extracted from unstructured texts (e.g. scenarios and/or support documents). Structure can be represented by linking and/or clustering relevant information.
Measurement indicators: Available functionality.

Requirement ID: SOTA-7 **Requirement Type:** Functional **Priority:** Should-have
Name: Memos
Description: To support recording comments, remarks, explanations and questions to text passages, concepts, knowledge structures. Ensuring linking enabling represent mutual relations and memberships.
Measurement indicators: Available functionality.

Requirement ID: SOTA-8 **Requirement Type:** Functional **Priority:** Must-have
Name: Non-RETE rule engine
Description: In order to use the rule engine in social simulations, the engine must be reasonably fast and therefore cannot frequently recompile its rule base. Therefore the RETE algorithm is inappropriate as a mechanism on which the rule base is constructed.
Measurement indicators: Rule engine not based on a RETE like algorithm available.

3.5.2. Tools preselection

Tool(s) selection for e-participation

Three software categories have been identified as categories worth considering for SOTA as they were identified as valuable and supportive for fulfilling the collaborative purposes of OCOPOMO – CMSs, e-participation platforms and wikis. These categories have been compared and it was argued and reasoned that available open source CMSs meet the objectives and fulfil the collaborative purposes of OCOPOMO better. Hence, the CMS category was selected for further investigation.

Seven CMS tools have been selected (1 tool written n JAVA, 5 tools written in PHP and 1 tool written in PYTON programming language) and evaluated in accordance with a set of criteria based on collected user requirements. The evaluation has revealed that all the tools under consideration are, in general, able to support the collaborative purposes of OCOPOMO. Finally, considering the evaluation results not only in the CMS category but across all categories, we have selected Alfresco CMS (with Alfresco Share collaboration platform) due to the following facts:

- It is the only one Java-based solution from the evaluated systems – in order to have tight integration of OCOPOMO platform, we have to find the best solution together with scenario analysis, simulation and rule engine tools. In this moment it is clear that other Java-based systems will be used (DRAMS, Repast), therefore it will be better for implementation tasks to provide the same platform for content and collaboration core of the system.
- Alfresco has a strong content repository without real opponent among the evaluated systems. This is important according to needs for management of complex and knowledge-intensive data objects (concepts, linking of objects, etc.) created in OCOPOMO during scenario analysis and policy modelling processes.
- Standards and novel technologies for content repositories like CMIS standard are available there and are reusable in many different ways and provide strong interoperability and integration possibilities. Alfresco is one of the leaders on standardization of content management paradigm.
- Maturity level is very good and combined commercial/community model with strong support of several leading companies in area predicts good sustainability, which could be a problem of purely open-source projects.
- Only one required functionality is not supported – polling, which is quite simple for implementation. If not considering polling in the evaluation, Alfresco is one of the best solutions. Integration possibilities, platform and standards behind the Alfresco and its repository are more important. Thus, it reflects all relevant implementation aspects for the project better than other solutions.
- Alfresco offers also good products for developers. It is possible to use free Alfresco SDK to customize it, prepare specific workflow, user interface tabs, portlets (called dashlets), all within the prepared development kit.

Tool(s) selection for scenario generation

For scenario generation two software categories were identified as valuable and supportive for fulfilling the collaborative scenario generation purposes of OCOPOMO – GSS and CWT.

GSS have been investigated and it was argued and reasoned that they are rather not applicable for fulfilling the collaborative scenario generation requirements in a way as intended in OCOPOMO. Therefore this category was not further considered.

CWT were selected for further investigation and ten tools have been selected and evaluated. The evaluation according to criteria based on the collected user requirements has shown that the non-proprietary CWT are, in general, able to support the collaborative scenario generation purposes of OCOPOMO. But it was argued, that from user perspective the similar functionality can be provided by the CMS category as well (moreover, this category benefits from the fact that prospective users are probably more familiar with tools from this category and therefore this category is more likely to succeed in attracting people to participate). Considering this as well as trying to minimise the number of tools to be integrated within the OCOPOMO ICT toolbox, it has been decided to prefer usage of CMS and therefore not to select any additional CWT tool but to utilise the already selected Alfresco CMS for scenario generation.

Tool(s) selection for scenario analysis

For scenario analysis three software categories – CAQDAS, argument visualisation tools and discourse analysis tools – were identified as valuable and supportive for fulfilling the collaborative scenario analysis needs of OCOPOMO.

Argument visualisation tools and discourse analysis tools have been investigated and it was argued and reasoned that they are rather not applicable for fulfilling the overall scenario analysis needs of OCOPOMO. These tools can only be used for additional support but not to perform the core tasks. Moreover, their methodological context is different from the one adopted by the project and their adoption for our purposes would be too demanding and outside of available resources. Because of this further consideration was not recommended.

CAQDAS were selected for further investigation and eight tools have been compared and evaluated. The evaluation according to the requirements has shown that the CAQDAS tools are in generally able to support the scenario analysis needs of OCOPOMO. A preselection of only non-proprietary CAQDAS tools took place. However, after better understanding of processes behind transformation of scenarios into formal models and definition of knowledge structures to be extracted from scenarios, all the evaluated CAQDAS tools have been rejected as considerable modifications for meeting the requirements of the scenario analysis as well as for a smooth transformation process are necessary. Hence, the development of a new analysis tool was recommended (being under our own control and therefore easily adoptable to current and potential upcoming requirements) and therefore no existing CAQDAS tool has been selected.

Tool(s) selection for formal modelling

Two categories of software tools have been identified as supportive for formal modelling task – ABM platforms and rule-based systems. In contrast to the above given areas (e-participation, scenario generation and analysis), the selected categories are not contradictory – both categories should be combined to provide suitable simulation software.

Five ABM platforms have been identified and investigated and based on this investigation two of them have been identified as attractive and viable candidates. In order to ensure sustainability and minimise expected problems, we prefer a candidate with longer history and reputation, wider user base and wider range of applications. The selected candidate is Repast simulation platform.

Three mature rule-engines have been identified. Unfortunately, it was argued and reasoned that although they are able to provide required functionality they are not applicable from the point of non-functional properties – since they are based on frequent recompilations of knowledge bases (using a variant of RETE algorithm), they are inappropriate for social simulations. Therefore no rule engine has been selected to be reused. Instead of this, a decision to develop a new rule engine not based on the RETE algorithm was taken. This new tool – a declarative rule-based agent modelling system (DRAMS) is currently under development within the workpackage WP5 [Moss et al., 2010].

To summarise the decisions taken, based on the analysis of state of the art while considering collected user requirements and current level of understanding of OCOPOMO processes, the following software tools have been selected to be used:

- Alfresco/Alfresco Share CMS (collaboration and scenario generation)
- Repast (agent-based simulation platform)

- DRAMS (rule engine - currently under development within the project’s workpackage WP5)

Since all the selected tools use Java technology platform, Java technologies (e.g. JDO, JMS, JCR, etc. based on JSR standards and specifications) are expected to be used to integrate the tools into the final OCOPOMO ICT toolkit.

Functionality which is required but not provided by the selected tools will be provided by adapting/enriching the selected tools or by additional tool(s) to be developed within next project phases.

4. ARCHITECTURE DESIGN METHODOLOGY

To produce an architecture of a system means to provide answers on many different questions. A desire to present them in the form of a single monolithic model often yields in a model which is rather complex and hard to understand. The reason is that such model must amalgamate information about different important aspects of the architecture which are of interest for different stakeholder categories. As a result, users are served poorly with such model since they may have problems to filter out information relevant for them and not to be distracted by their effort to understand those parts of the architecture which are valuable for other stakeholders.

One successful approach to manage the complexity of system architecture production is to employ a "divide and conquer" principle – to partition the problem and to attack it from different directions simultaneously. The architecture can be divided into several (interrelated) parts. Each of them deals with a particular aspect of the architecture. IEEE Standard 1471 [IEEE, 2000] has formalised the concepts behind this approach and provided a standardisation of terminology. The most important concepts are *view* and *viewpoint*, which are defined as follows:

- A *viewpoint* is a collection of patterns, templates and conventions for constructing one type of view. It defines the stakeholders whose concerns are reflected in the viewpoint, guidelines and principles and template models for constructing its views.
- A *view* is a representation of one or more aspects of an architecture, from the perspective of one or more concerns which are held by one or more of its stakeholders.

Thus, the architectural description of a system is composed from a set of views. Each view describes one particular aspect of the architecture of the system. This separation allows concentrating on the particular aspect only. That enables to conquer the complexity of a model representing this view, to select and use means the most suitable for the aspect being processed, and to communicate ideas and architectural decisions more clearly.

To guide architects in creating different views, viewpoints have been defined. A viewpoint represents "a library" for architects – a set of templates and patterns that can be used off the shelf to create a view. Thus, each viewpoint provides a guide how to deal with some aspect(s) of the architecture description – it can be used as a point of departure for producing one or more views which will be included into the description of the system architecture.

Viewpoints are independent of one another since viewpoints are defined in a way to be as disjoint as possible. But there can be architectural decisions which have impact on many or all views derived from the viewpoints. These decisions are usually driven by the need to ensure a certain quality property to be exhibited by the system. Such quality properties cannot be defined as an additional viewpoint, but must be addressed by several existing viewpoints. Such quality properties (often called non-functional properties) can be addressed by *architectural perspectives* which are defined as follows:

> An *architectural perspective* is a collection of activities, tactics, and guidelines that are used to ensure that a system exhibits a particular set of closely related quality properties that require consideration across a number of the system's architectural views.

Perspectives are orthogonal to viewpoints – they can be applied to views. Such application ensures that a perspective's system-wide quality property is addressed within a view. In principle, every perspective can be applied to every view. In practice, a perspective is applied to only some views which are relevant to the perspective in a particular context – not all combinations of perspectives and views are needed.

In order to illustrate the used concepts, Figure 4 provides an overall picture of these concepts and their relationships [Rozanski and Woods, 2005].

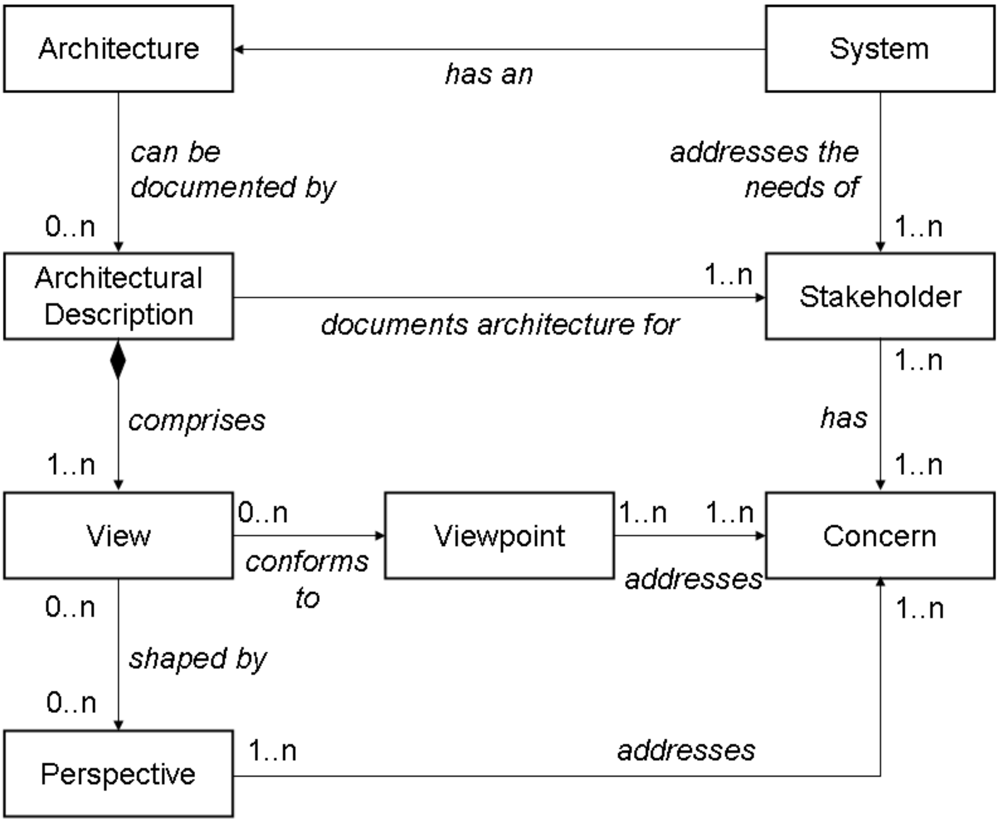


Figure 4 Interrelations between core architecture concepts

- A *system* is built in order to address the needs, concerns, goals, and objectives of its stakeholders.
- A *stakeholder* is a person, group, or entity with an interest in or concerns about the realisation of the architecture.
- A *concern* about an architecture is a requirement, an objective, an intention, or an aspiration a stakeholder has for the system.
- An *architecture* of a system is a particular arrangement of static and dynamic structures that has the potential to exhibit the system's required externally visible behaviours and quality properties.
- An *architectural description* documents the architecture of the system. It consists of a set of views addressing concerns of system's stakeholders.
- A *viewpoint* defines the aims, intended audience, and content of a class of views and defines the concerns that views of this class will address.
- A *view* conforms to a viewpoint and so communicates the resolution of a number of concerns. The content of a view can be shaped by a number of perspectives.
- A *perspective* addresses a number of concerns of the system's stakeholders in order for the system to exhibit the quality properties considered by that perspective.

According to [Rozanski and Woods, 2005], there are six core viewpoints which are applicable for traditional information systems. These viewpoints are complemented by ten perspectives which are applicable to these viewpoints. Not all viewpoints and perspectives are applicable for describing the architecture of any system. Depending on a particular case, some of them are more relevant than the others. On the other hand, the architecture of a particular system can require the addition of a new viewpoint/perspective extending the available set.

In order to describe the architecture of the OCOPOMO system, only a few basic views and perspectives have been selected:

- **Functional view** - describes the system's functional elements, their responsibilities and their interactions.
- **Information view** - describes the way in which the architecture stores, manipulates, and distributes information.

The following perspectives have been selected:

- **Internationalisation perspective** - the ability of the system to be independent from any particular language and/or country.
- **Interaction perspective** – describes user interface views and possible interactions with the system which are available to users.
- **Usability perspective** - the ability of the system to enable users to interact with the system easily and effectively.

The selection was based on characteristics of the prospective OCOPOMO system, on the actual phase of development represented by this report and on the structure of the OCOPOMO project. As a result, several view/perspectives have been disqualified from the selection. Some of them focus on aspects considered within other parts of the project (e.g. Development view), aspects related with a very detailed point of view (e.g. Performance perspective), aspects relevant only for more mature state of the project (e.g. Operational or Deployment views) or aspects not considered relevant (e.g. Location perspective).

5. USER-ORIENTED PROCESS PERSPECTIVE

The OCOPOMO approach is based on complex and interrelated processes which require careful coordination of many different users of the system (i.e. stakeholders, policy modellers, policy analysts, etc.) who communicate and collaborate on diverse tasks during the process of policy modelling. Our goal is to create an ICT toolbox supporting the OCOPOMO approach and ensure robustness of the platform and avoidance of possible defects (e.g. task duplication, high efforts of data search and limited functionality of available tools), which can hinder workflow of policy process modelling and result in low satisfaction of stakeholders.

Taking into account aforementioned, we believe that particularly in the policy making area the stakeholders' view of the process is very important. The satisfaction of users of the ICT toolbox influences the outcome of their work, namely the quality and accuracy of scenarios, policy models and simulation results as well as policy decisions in the end. In order to meet the goal we have decided to create use case diagrams in Unified Modelling Language (UML) and their descriptions, depicting a user-oriented view of the system, that show system functions and roles of participating actors.

The developed use case diagrams correspond with the revised list of requirements provided in deliverable D1.1 (Stakeholder Identification and Requirements for Toolbox, Scenario Process and Policy Modelling) [Bicking et al., 2010]. The analysis of the OCOPOMO approach revealed that the process to be supported by the ICT toolkit can be divided into the following main areas:

1. registration/login,
2. initiation,
3. working with the project,
4. collaboration,
5. scenario generation,
6. scenario analysis,
    a. quantitative data analysis,
    b. qualitative data analysis,
        i. extraction of phrases from natural language descriptions,
        ii. issue generation, generation of relations and relation clusters,
        iii. expertise-based relation,
    c. network visualisation,
7. policy modelling,
8. simulation,
9. validation and evaluation.

The following subsections demonstrate use cases in each of these recognised areas of actors' interaction with the system.

5.1. REGISTRATION

The registration and login procedure is presented in the following figure. To take part in the project invited User needs to register by providing the following information about him/her: User name, Email, Password, Code to avoid spam bots, and to accept general Terms and Conditions. Uninvited user who wants to take part in the project can send request for invitation with a description of himself/herself and his/her motivation. A review and approval process is needed in that case.

Once a user registers he/she will be invited to provide a personal profile, which includes: his/her personal information and contact details, topics of interest to him/her (in order to be notified about new topics posted on the forum related to that processes), newsletter registration[125]. The user profile information can be modified at any time by the profile owner. If a registered user wants to delete his/her profile and stop being a registered member, he/she must/can do this in the system.

Only Users who accept invitation sent by Facilitator can register. Accepting an invitation means clicking the link added to the e-mail with the invitation.

After the initial registration, members can login each time they wish to access the site by providing their user name or email and password. In case of forgetting the password User can use password reminder.

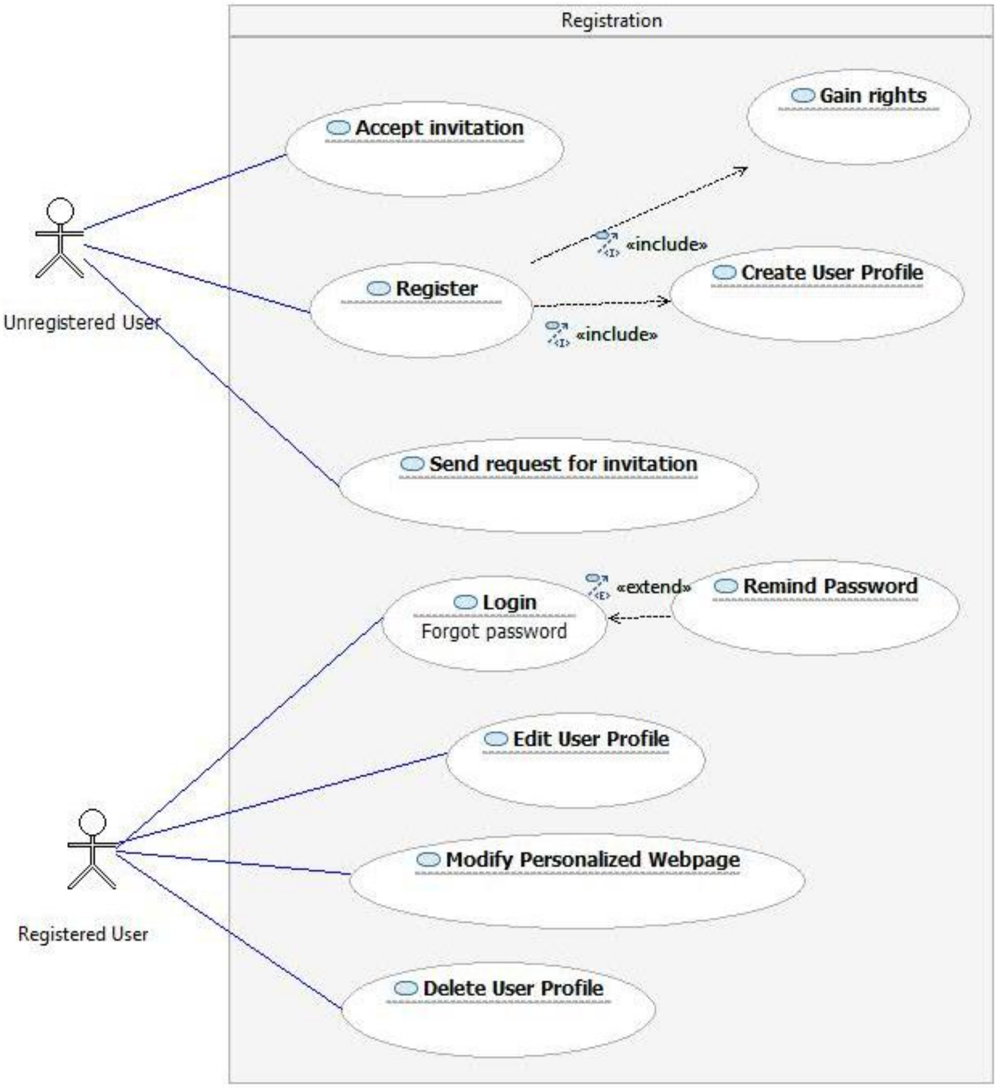


Figure 5 Use case "Registration"

[125] See also the user requirement T-30 as well as the definition of newsletter data object in section on Information view.

The system shall provide a personalized webpage for registered and logged in users, which is customized according to the user's preferences. This means that the user 1) can choose which information should be visible (events, news, forums, etc.) at which place on the webpage, and 2) the interesting information is highlighted. The user has to be registered and logged in to see the personalized webpage. The personalized overview does not replace the start page.

Use Case Diagram Name	Registration/login
Related Requirements	I-F-I1, I-F-I2, I-F-I3, I-F-I4, I-F-I5, I-F-I6
Goal in Context	The registration/login area enables Unregistered User to register and create a profile. After completing the registration User can login (use password reminder if needed), edit or delete profile and personalize webpage.
Preconditions	The user has to be invited.
Successful End Condition	The registration/login area is viewed and ready to be edited.
Failed End Condition	The registration/login area is not rendered.
Primary Actors	Unregistered User, Registered User
Trigger	The User initiates registration or login.
Main Flow	**Step. Action**
	1. The invited User accepts the invitation. 2. The User can register. 2.1. *Include: Create User profile.* 2.2. *Include: Gain Rights.* 3. The uninvited User can send request for invitation. 4. Registered User can login on her/his profile by providing his/her user name or email and password. 4.1. *Extend:* Remind Password. User can use password reminder. 5. User can customize personalized webpage according to her/his preferences. 6. User can edit his/her profile if needed. 7. User can delete his/her profile and stop being a registered member.

Table 18 Description of the "Registration" use case

5.2. INITIATION

The use case diagram (Figure 6) illustrates the initiation of the project which includes the creation of a collaboration space and the generation of the policy description. While creating the policy description initiator (facilitator) develops text descriptions and uploads background documents referring to the policy case. After this, the initiator (facilitator) invites users (both registered and unregistered users, i.e. users who are not yet involved but who are identified as valuable contributors) and assign them rights.

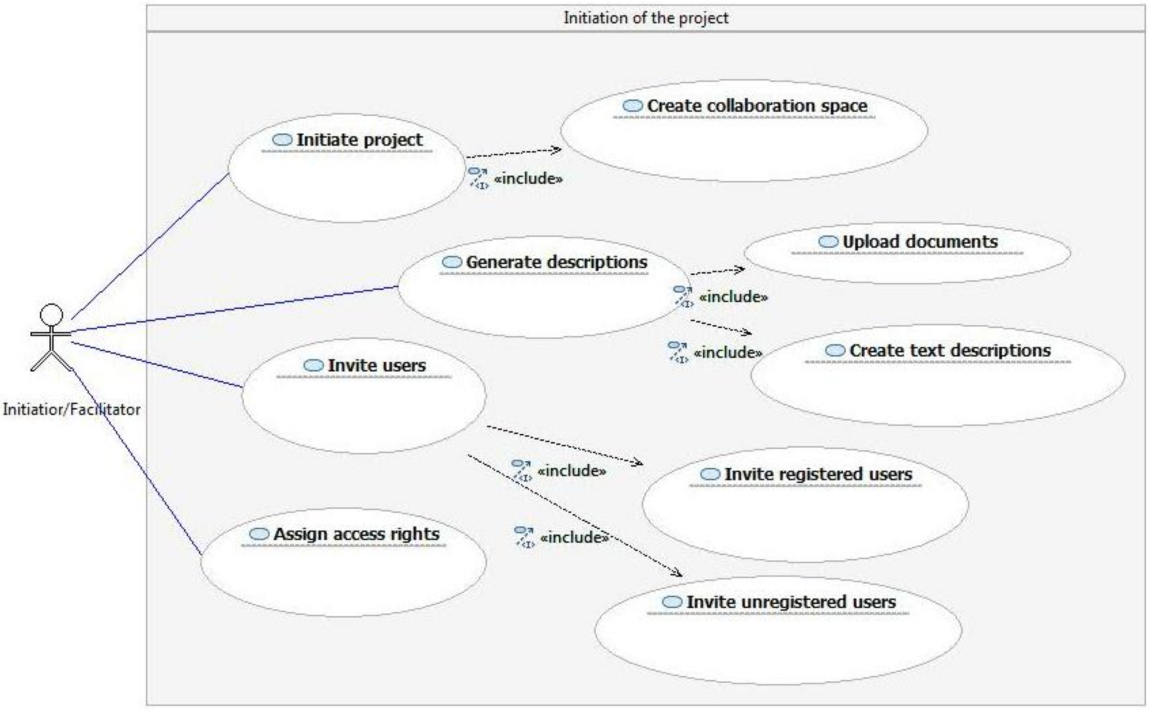


Figure 6 Use case "Initiation of the Project"

Use Case Diagram Name	Initiation of the project
Related Requirements	T-5
Goal in Context	The Facilitator initiates the project by generation of the collaboration space and policy description (upload of referring documents and description of the policy case). The initiator/facilitator invites users (registered and unregistered users) and assigns them access.
Preconditions	1. The Facilitator must be registered at the system. 2. The Facilitator must be logged in.
Successful End Condition	The project initiation is completed and users have access to the collaboration space and documents.
Failed End Condition	The project initiation cannot be finished or Users do not have access to the collaboration space.
Primary Actors	Facilitator / Initiator
Trigger	The Facilitator initiates new project.
Main Flow	**Step. Action**
	1. The Facilitator initiates a new project by selecting this functionality in the system (he/she is automatically given all rights). 1.1. *Include: Create collaboration space.* The system/platform creates a unique collaboration space for the project automatically when the Facilitator initiates a project. 2. The Facilitator generates description, including: 2.1. *Include: Create text description.* The Facilitator can create text description of the project. This description includes

	the creation of the following information: 2.1.1. Name: Name of the project e.g. “Open collaboration for policy modelling” 2.1.2. Abbreviation: Abbreviation of the project for easier communication e.g. “OCOPOMO” 2.1.3. Description: Textual description of the project. The system provides a form which supports the Facilitator in providing this information. 2.2. *Include: Upload documents.* The Facilitator uploads corresponding documents. 3. The Facilitator invites users, which will be allowed to view the project: 3.1. *Include: Invite registered users.* In order to invite registered users, the Facilitator can select users from a list. It is also possible to search for users by name and/or e-mail. 3.2. *Include: Invite unregistered users.* The Facilitator can invite new users to cooperation. Therefore the Facilitator can add e-mail addresses to a list of to be invited users. After confirming the “invitation”, the system sends an e-mail to each user proposed by the Facilitator. The e-mail contains an automatically generated text describing OCOPOMO and the project initiated by the Facilitator. In addition, the e-mail contains a link, which directly leads the user to the project either to a log-in or to registration web page. After log-in the user is directed to the collaboration space of the project. 3.3. The Facilitator assigns particular rights to the users. When inviting particular users (registered as well as unregistered) the Facilitator can already assign further access rights to them (i.e. write, invite).

Table 19 Description of the “Initiation of the Project” use case

After completing the initiation process users gain access to the collaboration space, which will be customized for the project within the platform.

5.3. WORKING WITH THE PROJECT

The next use case diagram (Figure 7) illustrates the process of working with the project and it describes those activities, which are possible with a project.

The Facilitator/Authorised User can update the initial description of the project by uploading and removing documents as well as revising initial text description. Moreover, he/she can invite additional users and assign them rights. The Facilitator and Authorised User can change access rights of users at any time.

The registered users can only update the existing description of policy case.

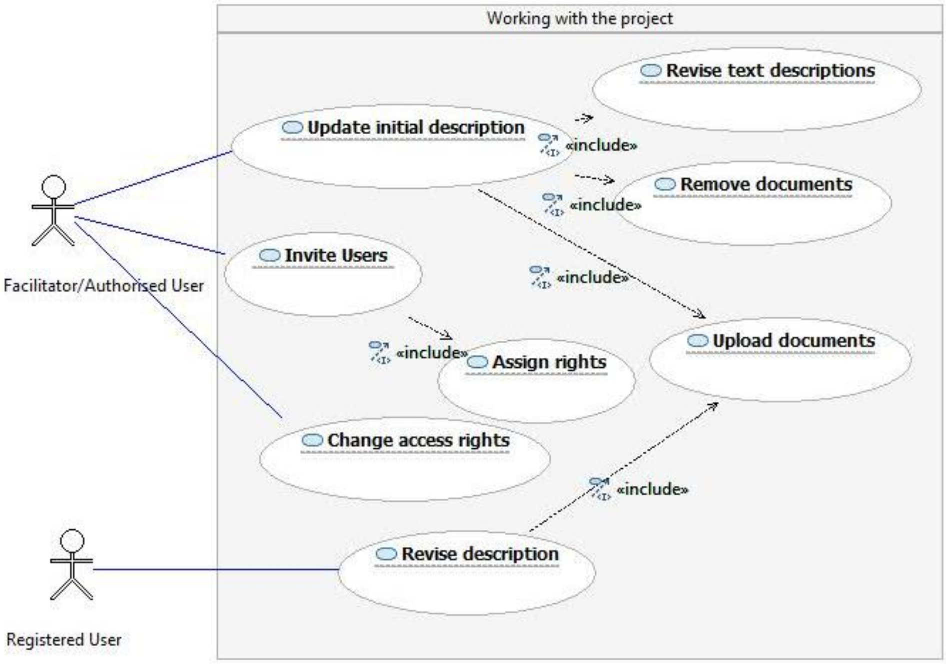


Figure 7 Use case "Working with the Project"

Use Case Diagram Name	Working with the project
Related Requirements	*[126]
Goal in Context	The use case starts when a project has been initiated by Facilitator. The Facilitator and Authorised Users are able to update the descriptions and to add new ones. Moreover he/she can invite and assign access rights to users. Registered users can review and update documents and description provided by Facilitator.
Preconditions	1. The Facilitator and User must be registered at the system. 2. The Facilitator must be logged in. 3. The User receives the view of the data according to the rights he/she owns. 4. The User can work on the data according to the rights he/she owns.
Successful End Condition	Facilitator and Authorised Users could manage the users and their rights as well as run the project (revise the description).
Failed End Condition	The description cannot be changed, users cannot be invited or rights cannot be changed/allocated.
Primary Actors	Facilitator/Authorised User, Registered User
Trigger	The User or Facilitator/Authorised user works with a project.
Main Flow	**Step. Action**
	1. The Facilitator or authorised users can update initial description, including:

[126] If a use case (here as well as below) is not backed up by already defined user requirements, then at least one new user requirement will be defined.

	1.1 *Include: Revise text description.* The Facilitator can revise text description of the project. This description includes the creation of the following information: a. Name: Name of the project e.g. “Open collaboration for policy modelling” b. Abbreviation: Abbreviation of the project for easier communication e.g. “OCOPOMO” c. Description: Textual description of the project. d. The system provides a form which supports the Facilitator in providing this information. 1.2 *Include: Upload document.* The Facilitator uploads additional documents. 1.3 *Include: Remove document.* The Facilitator removes selected documents. 2. The Facilitator and authorised Users can invite users, which are allowed to view/edit the project. 3. The Facilitator and authorised users are able to change access rights. 4. The Facilitator and authorised users are able to assign access rights. 5. The Registered User, after invitation, can revise description of the project. 5.1 *Include: Upload document.* The Registered User uploads additional documents.

Table 20 Description of the “Working with the Project” use case

5.4. COLLABORATION

The collaboration space is available on-line both for facilitators and end users (Figure 8). The role of the ICT toolbox will be essential for supporting contextual social knowledge exchange and seamless interaction within a complex virtualized world, where users are in the foreground, at the centre of all attentions, while supporting technologies operate in the background, almost invisible.

Registered users are able, for example, to chat, use discussion forums, use help, search, ask facilitator, use annotation system and change management option, create scenario, upload documents, open simulation results, use opinion polling, view calendar and newsletter. The authorised user, on the other hand, can create events in the calendar and publish newsletter.

The main features of collaboration space are depicted below.

Search engine

The search engine for facilitators shall help the facilitator to differentiate between relevant and irrelevant documents/inputs (to find those which are relevant) and to show the status of the analysis of the document (e.g. completed, partly completed and not yet started). Both users and facilitators can look for a specific topic. A template of metadata will be provided to characterize the documents. The metadata will then be used for searching for specific documents and contents.

Calendar

The calendar shows events related to the project in a well-arranged form on yearly, monthly, weekly and daily base. Responsible users (with granted access rights) can enter an event to the calendar. The calendar should have a function of sending a reminder about the event to all predefined users or groups of users.

Change notification option

Change notification option informs the User about the changes in the project according to the rights he/she has and his/her settings in profile (i.e. depending on the role the User has in the project).

RSS

All users can install RSS reader, to be able to check for new information, tasks, downloads on a regular basis (e.g. during an unsynchronised scenario generation session) using system feed.

E-mail notification system

E-Mail notification system should provide an awareness mechanism (daily/weekly/monthly) allowing participants to be informed on newly published or modified content in discussions, CMS, etc.

Opinion polling

Opinion polling tool should support definition of different types of questions and answers – e.g. multi-choice questions, text-based inputs (answers), selection of a specific part on a map, selection of text parts (fragments) in a text, etc. Authorised users are able to conduct an opinion poll and define the users who are allowed to participate or to organize an open poll. Users can participate in the opinion polls and they can change their opinion - answers provided before, i.e. they can produce a new version of the filled in form. It is possible to support launching/closing the opinion polling according to the defined setting (e.g. time interval, the number of participants, percentage of the filled in forms from the whole group, etc). The opinion polling tool is able to produce a graphical output from the survey results (using graphs, diagrams, etc.).

News

Authorized users can publish news and link them to other parts of the system. News is readable by all users (no need to login). The news feature shall provide an overview about recent published news with date and title, last modified elements notification, etc.

Discussion forum

Discussion forums support both moderated and non-moderated discussions. In the first case authorized users (e.g. facilitator in case of the scenario building) can moderate a discussion within the discussion forum. Contributions to the forum will be automatically published and the moderator is informed of the new contributions. The moderator can decide to withdraw a contribution. The forum will be used as a consultation tool to ask users about their opinions on specific issues. The discussion forum is applied to enable a formal discourse on topics of interest extracted from the scenario in order to advance it and to provide specific information on it. Within well-directed moderated discussion forums stakeholders are consulted to express their opinions, recommendations and concerns regarding completeness and assessment of desk research results. Discussion forums help to relate and advance descriptions while stakeholders are discussing their opinions with other stakeholders. The system shall publish rules for comments' moderation at the "Rules for engagement" section in order to avoid the accusations of censorship.

It is possible to organize discussion threads within forums in different types of order like chronological or topic-based (hierarchical structure of threads, topics, and messages). In more details, scenario building requires that contributions can be depicted structured through topics (i.e. several discussions are possible at the same time concerning different topics of interest extracted from the scenario) or chronologically (i.e. discussions to one topic should be in a chronological order).

In case of non-moderated discussions system shall publish the written comments automatically, although the content administrator will be able to erase them at any moment.

Users are able to attach a relevance feedback to contributions in the discussion forum using a rating scale (e.g. 2 – strongly agree, 1 – agree, 0 – neutral, -1 – disagree, -2 – strongly disagree) about the content.

Chat
The chat is a feature in the collaboration space, which provides a text-based group chat embedded into the collaboration space. The chat can be used by the users in order to discuss the project. Users can use it without further log-in; they just need to select the functionality and can start writing. On one hand, users have the chance to meet other users. On the other hand, the facilitator can arrange a time and invite users to participate in a chat. To invite users, the facilitator can send a newsletter to the project members.

Comment
Authorized users (e.g. facilitator in case of the scenario generation) can decide whether the content in the system can be commented upon. Commenting should have always the same style, does not matter what is commented. Users are able to comment most of the sources within the system.

Newsletter
Responsible users (with granted access rights) can create (publish) a newsletter and send it to the subscribed users.

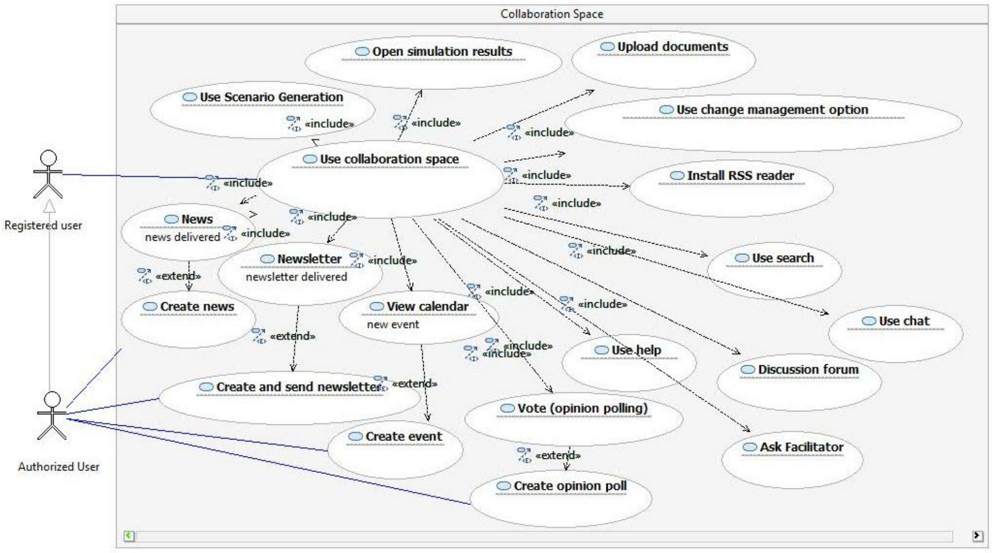


Figure 8 Use case "Collaboration Space"

Use Case Diagram Name	Collaboration space
Related Requirements	T-1, T-1-1, T-1-2, T-1-3, T-1-4, T-1-5, T-4; T-5, T-7, T-8, T-9, T-10, T-11, T-12, T-14, T-24, T-28, T-29, T-30, T-34, I-1, I-5, I-6, I-7, I-32
Goal in Context	The collaboration space allows the User to fully participate in the project and facilitate the collaboration among Users providing tools like calendar, newsletter, help, search tool, chat, discussion forum, communication with Facilitator, opinion polling, annotation feature and change management option and RSS. The collaboration space gives access to all documents, scenario generation tool and results of simulations. The Authorized User can add member, add event

	and publish newsletter.
Preconditions	The User has to be authenticated and has particular rights.
Successful End Condition	The “collaboration space” allows the collaboration among users within a project.
Failed End Condition	The collaboration space elements are not rendered.
Primary Actors	Registered User, Authorized User
Trigger	The User opens collaboration space.
Main Flow	**Step. Action**
	1. The Registered User uses collaboration space during the project, including: 1.1. *Include: Use chat*. The User can use a chat to communicate with other users of the platform. 1.2. *Include: Discussion Forum*. The User can use forum to communicate with other users of the platform. 1.3. *Include: Use help*. The User can use help and assistance feature from every webpage of the system. 1.4. *Include: Ask Facilitator*. The User can ask facilitator about the project or the system, etc. 1.5. *Include: Use search*. The user has access to search tool to search text descriptions and uploaded documents. 1.6. *Include: Install RSS reader*. The User can set up RSS to be able to check for new information and uploaded documents. 1.7. *Include: Use change management option*. The User is informed about the changes within the project according to the rights he/she has and his/her settings in profile (i.e. depending on the role the User has in the project). 1.8. *Include: Vote (opinion polling)*. The User can take part in opinion polls. 1.8.1.*Extend: Create opinion poll*. The authorised User can create opinion poll. 1.9. *Include: Upload documents*. The User can upload documents related to the project to database. 1.10. *Include: Open simulation results*. The User can check project simulation results. 1.11. *Include: Use Scenario Generation*. The User can open Scenario Generation tool. 1.12. *Include: View Calendar*. The User can check calendar of project events. 1.12.1. *Extend: Create events*. The Authorized User can add new events to calendar. 1.13. *Include: Newsletter*. The newsletter is delivered to the User by e-mail. 1.13.1. *Extend: Create and send Newsletter*. The Authorized User can create newsletter and send it to the collaborating user group. 1.14. *Include: News*. The news is delivered to the User on-line. 1.14.1. *Extend: Create News*. The Authorized User can create news which is delivered to the collaborating users. 2. The Authorized User can add new events to the calendar. 3. The Authorized User can create newsletter and send it to the

	collaborating user group. 4. The Authorised User can create news which is delivered to collaborating users. 5. The Authorised User can create opinion poll.

Table 21 Description of the “Collaboration Space” use case

5.5. SCENARIO GENERATION

In order to represent different views of stakeholders in a sensible way the users (authorised for this activity) will generate scenarios. The initial scenario will be created by the initiator of a policy (user authorised to initiate policy) process and modeller who can start new iteration of scenario generation which will be published at the collaboration space (it will be opened for viewing and manipulation). Stakeholder users will be allowed to express their views on the policy case via either further elaboration on the initial scenario or by generating new (alternative) scenarios. When a scenario will be closed or reopen by the facilitator, then involved stakeholders will receive information about this action. Scenario extension and update shall include the opportunity to rate the scenario or parts of it, as well as to discuss on the scenario. For supporting the latter, an annotation feature is needed.

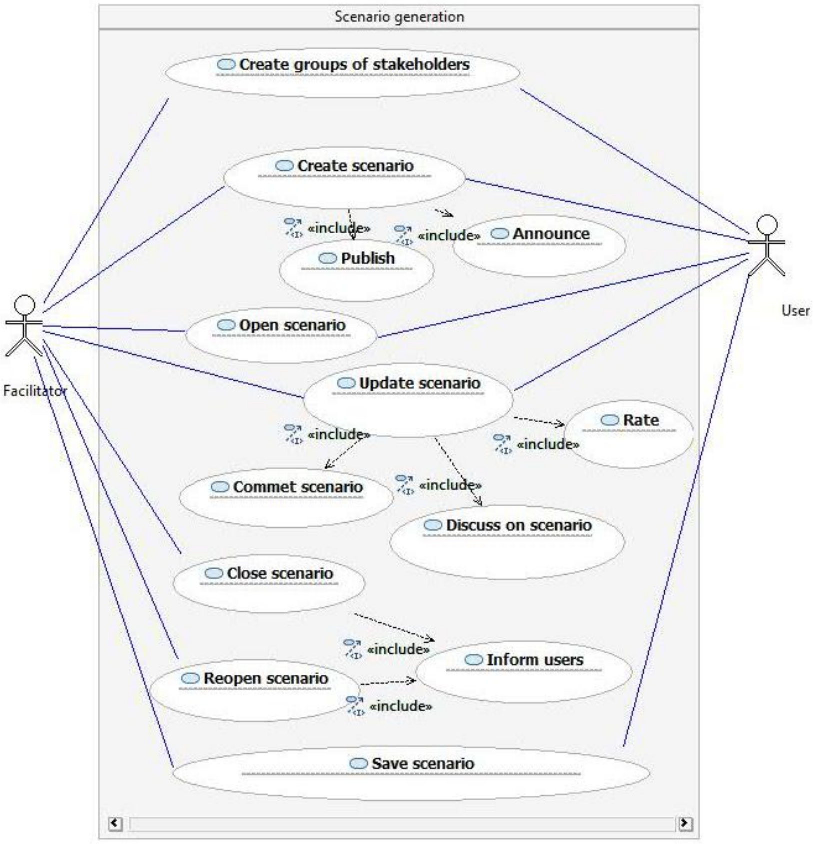


Figure 9 Use case “Scenario Generation”

Use Case Diagram Name	Scenario Generation
Related Requirements	I-2, I-3, I-4, I-10, I-11, I-13, I-14, I-15, I-19, I-22, T-25
Goal in Context	The scenario generation tool allows the Facilitator to manage scenario process (i.e. create scenario as well as update, close and reopen existing scenarios). Scenario generation tool allows the user (with appropriate access rights) to take part in scenario generation process (i.e. creating, viewing and updating scenario).
Preconditions	1. The facilitator / the user must be registered at the system 2. The facilitator / the user must be logged in 3. The facilitator / the user receives only the view of the data according to the rights he/she owns 4. The facilitator / the user can only work on the scenarios according to the rights he/she owns • the Facilitator can create scenarios as well as update, close and reopen existing scenarios • the User can create, view and update scenarios
Successful End Condition	The collaboration space allows generating scenarios by the user and managing the scenario generation process by the facilitator.
Failed End Condition	The collaboration space does not render the required features/services for collaborative scenario building (writing and discussing)
Primary Actors	Facilitator, User
Trigger	The facilitator publishes the initial scenario and invites the users for contributions thereby starting the scenario generation process.
Main Flow	**Step. Action**
	1. The Facilitator (or the User) can create new scenario, including: 1.1 *Include: Publish.* The Facilitator publishes new scenario. 1.2 *Include: Announce.* The Facilitator informs Users about creation of new scenario. 2. The Facilitator (or the User) is able to create groups of stakeholders to assure when needed group homogeneity in scenario generation process. 3. The Facilitator (or the User) can open current scenario. 4. The Facilitator (or the User) can update current scenario, including: 4.1 *Include: Rate*. The Facilitator and the Users can rate current scenario. 4.2 *Include: Discuss on scenario*. The Facilitator and the Users can discuss current version of scenario. 4.3 *Include: Comment scenario*. The Facilitator and the Users are able to comment on current scenario. The Facilitator manages scenario generation process: 5. The Facilitator can close current scenario; 5.1 *Include: Inform users.* The Facilitator informs Users about closing the scenario. 6. The Facilitator can reopen scenario; 6.1 *Include: Inform users.* The Facilitator informs Users about reopening scenario.

	7. The Facilitator and the Users can save current work during scenario generation process.

Table 22 Description of the "Scenario Generation" use case

5.6. SCENARIO ANALYSIS

The main activities of scenario analysis are summarized in related use case (see Figure 10). The detailed steps to be performed subsequently are described in the following subsections: qualitative data analysis, quantitative data analysis and network visualization. The export of the results of scenario analysis shall be possible in XML to feed the result into DRAMS. Hence, an analysis tool needs to provide the XML-Export.

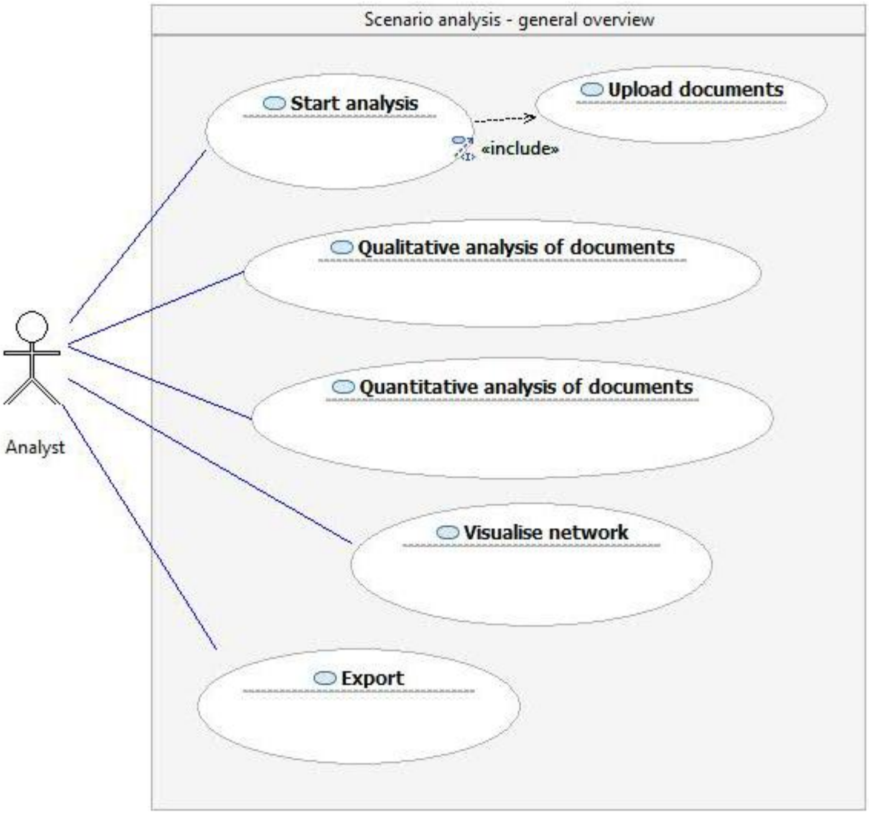

Scenario analysis - general overview
Start analysis
Upload documents
«include»
Qualitative analysis of documents
Quantitative analysis of documents
Visualise network
Export
Analyst

Figure 10 Use case "Scenario Analysis - General Overview"

Use Case Diagram Name	Scenario analysis – general overview
Related Requirements	T-39, T-40
Goal in Context	The scenario analysis is conducted by the CAQDAS and the CCD analysis tool.
Preconditions	1. The analyst must be registered at the system 2. The analyst must be logged in

	3. The analyst receives the view of the data according to the rights he/she owns 4. The analyst can only work on the data according to the rights he/she owns
Successful End Condition	The CAQDAS and the CCD are viewed and able to be used.
Failed End Condition	The CAQDAS and the CCD do not render the required features/services, i.e. it delivers useless input for the simulation model.
Primary Actors	Analyst
Trigger	The first evidence-based stakeholder-generated user scenarios are completed (available), so that the analyst can initiate the qualitative data analysis.
Main Flow	**Step. Action**
	1. The Analyst starts analysis of scenario, including: 1.1. *Include: Upload docs.* The Analyst can upload additional documents. 2. The Analyst enters quantitative analysis of documents. 3. The Analyst enters qualitative analysis of documents. 4. The Analyst enters network visualization in which all scenario descriptions are mapped onto a network depending on the query. 5. The Analyst exports the results of analysis to be imported into DRAMS.

Table 23 Description of the "Scenario Analysis - General Overview" use case

5.6.1. Qualitative analysis of documents

Scenario analysis is mainly qualitative data analysis as scenarios are based on narrative texts. As the process is complex and requires full understanding among OCOPOMO platform developers, we decided to present all its parts in the detailed subsections (see Figure 11): extraction of phrases, issue generation, generation of relations and relation clusters, expertise-based relations.

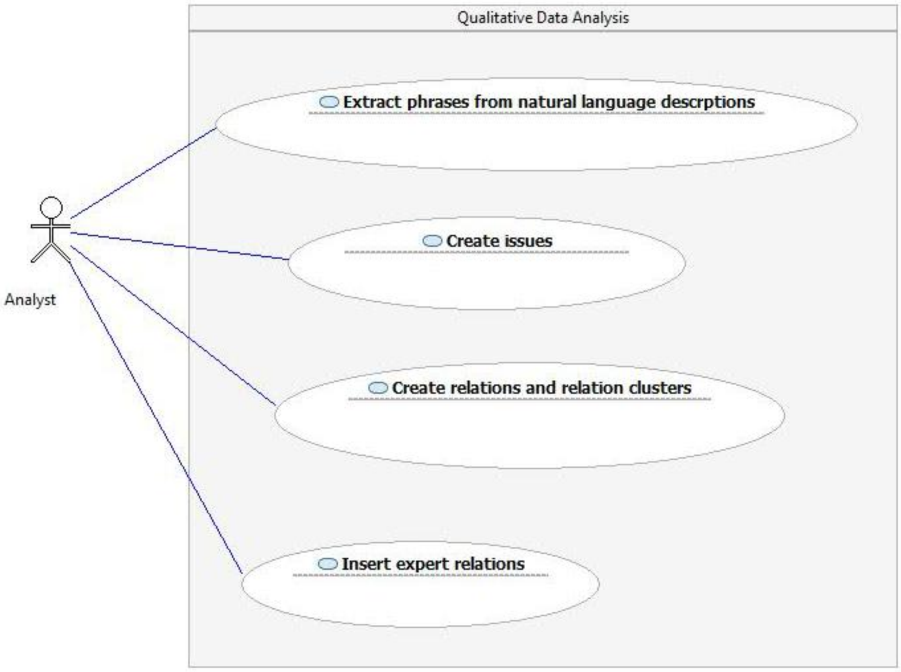

Qualitative Data Analysis
Extract phrases from natural language descrptions
Create issues
Create relations and relation clusters
Insert expert relations
Analyst

Figure 11 Use case "Qualitative Data Analysis"

Use Case Diagram Name	Qualitative Data Analysis (QDA) of natural language descriptions
Related Requirements	T-40
Goal in Context	QDA of natural language descriptions (i.e. background documents and evidence-based stakeholder-generated scenarios). CAQDAS provides features to search, structure, organize, categorize, and annotate textual data. The CAQDAS helps selecting and categorizing phrases as well as managing the corresponding metadata (i.e. insertion, revision and removal of metadata).
Preconditions	1. The analyst must be registered at the system 2. The analyst must be logged in 3. The analyst receives the view of the data according to the rights he/she owns 4. The analyst can work on the data according to the rights he/she owns
Successful End Condition	1. The analyst receives the view of the data according to the rights he/she owns in the CAQDAS/system 2. The analysts can work on the data according to the rights he/she owns in the CAQDAS/system 3. The CAQDAS provides the features necessary to enable analysts to analyse the qualitative data in order to deliver high quality input for the CCD and the subsequent simulations
Failed End Condition	The CAQDAS does not render the required features/services, i.e. it delivers useless input for the CCD and the simulation model.
Primary Actors	Analyst
Trigger	Three cases may trigger the QDA: 1. The first background documents for the policy case are

	available, so that the analyst can initiate the qualitative data analysis. 2. The first evidence-based stakeholder-generated user scenarios are completed (available), so that the analyst can initiate the qualitative data analysis. 3. The first simulation-based scenario is available, so that the analyst can initiate the qualitative data analysis.
Main Flow	**Step. Action**
	1. The Analyst can extract phrases from natural language descriptions 2. The Analyst can generate issues 3. The Analyst can generate relations and relation clusters 4. The Analyst can insert expert relations

Table 24 Description of the “Qualitative Data Analysis” use case

5.6.1.1. Extraction of phrases from natural language descriptions

The qualitative analysis of documents starts with the elaboration of the text material, i.e. a relevant text passage is highlighted, the text passage, called phrase, is processed and coded.

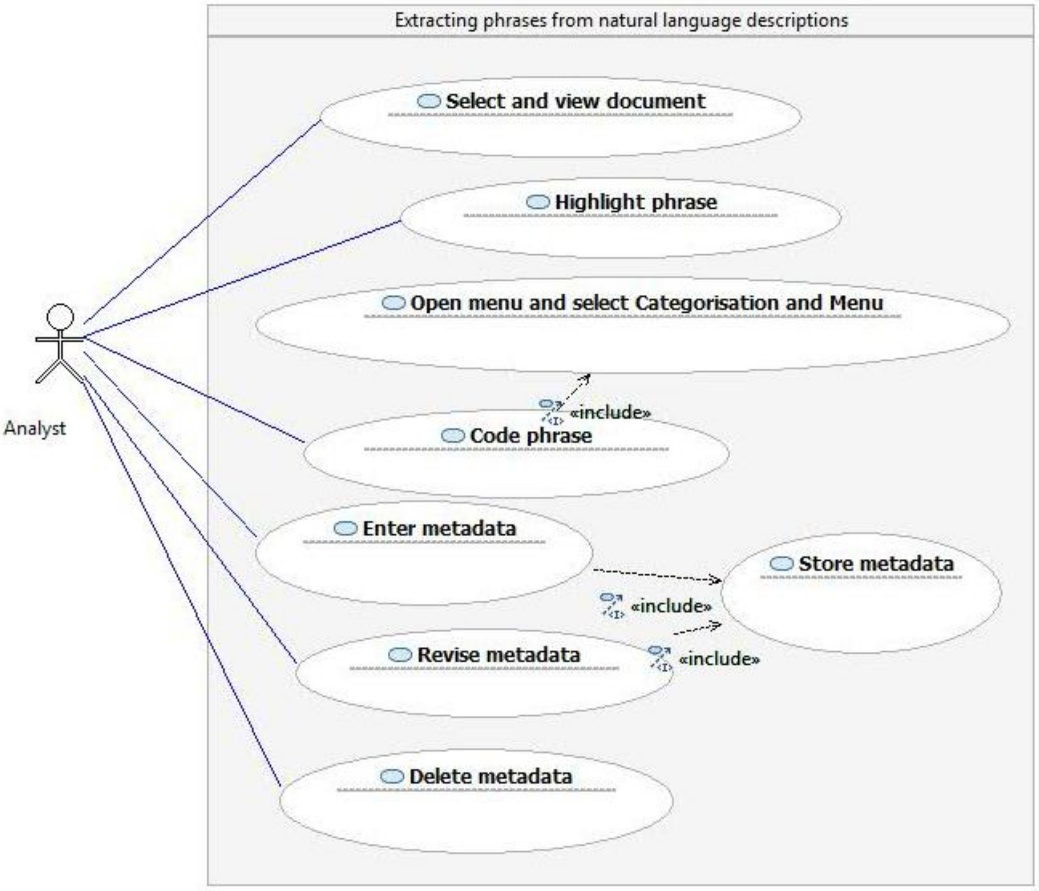


Figure 12 Use case “Extracting Phrases from Natural Language Descriptions”

To assure traceability the text material should be numbered consecutively by the line and paragraph to determine the position of the phrase extracted (it enables immediate localisation of important text passages and checking the broader context, in which an issue originally occurs). The code might be a single word or an acronym (single character or a combination of characters) or a combination of words.

The identified collection of text passages requires precise naming. Working within a team of people may lead to different understandings of text passages, which challenges readability. Hence, it is important that interim results are self-explanatory in order to facilitate a common understanding within the team of analysts.

All possible use cases that are referring to the extraction of phrases are depicted in Figure 12.

Use Case Diagram Name	Extracting phrases from natural language descriptions
Related Requirements	T-39
Goal in Context	The tool allows for selection of phrases from natural language descriptions.
Preconditions	1. The analyst must be registered at the system. 2. The analyst must be logged in. 3. The analyst receives the view of the data according to the rights he/she owns. 4. The analyst can work on the data according to the rights he/she owns.
Successful End Condition	The CAQDAS is viewed and able to be used.
Failed End Condition	The CAQDAS does not allow extracting, coding and defining phrases from natural language descriptions.
Primary Actors	Analyst
Trigger	The analyst found a relevant text passage; hence he/she initiates the extraction and coding of the phrase from natural language description.
Main Flow	**Step. Action**
	1. The Analyst selects and views a natural language description (i.e. either a background document or a scenario) 2. The Analyst goes through the text and highlights relevant text passages (i.e. phrases) 3. The Analyst codes the phrase, including: 3.1. *Include: Open menu and select Categorisation and Menu.* The Analyst can select phrase categorization and menu. The CAQDAS sends the respective view to the Analyst (i.e. opens a respective window to code the phrase) and automatically inserts a link from the document to the data base entry including the information where to find the original text passage in which document. 4. The Analyst enters metadata of analyzed document. 4.1. *Include: Store metadata.* The metadata such as title and description are stored.

	5. The Analyst can revise metadata of analyzed documents. 5.1. *Include: Store metadata.* The revised metadata are stored. 6. The Analyst can delete metadata.

Table 25 Description of the "Extracting Phrases from Natural Language Descriptions" use case

5.6.1.2. Issue generation

Subsequent step is the assignment of the phrases, identified in the different texts, to the right issue avoiding multiplication of new issues. In doing so, tabs ought to set up to define the metadata and to allow updating them. The title of the issue shall suggest the content and meaning of the issue, and be restricted to the meaning of the phrases included. Hence, coding grounds the interpretation of the analyst (i.e. codes cannot be generated automatically).

With a group of analysts it might be challenging to create a compromise on an issue title or abstract or any other characteristics. Thus, it is necessary to be able to flexibly revise or delete the metadata during the scenario analysis process.

All use cases related to issue generation are depicted in Figure 13.

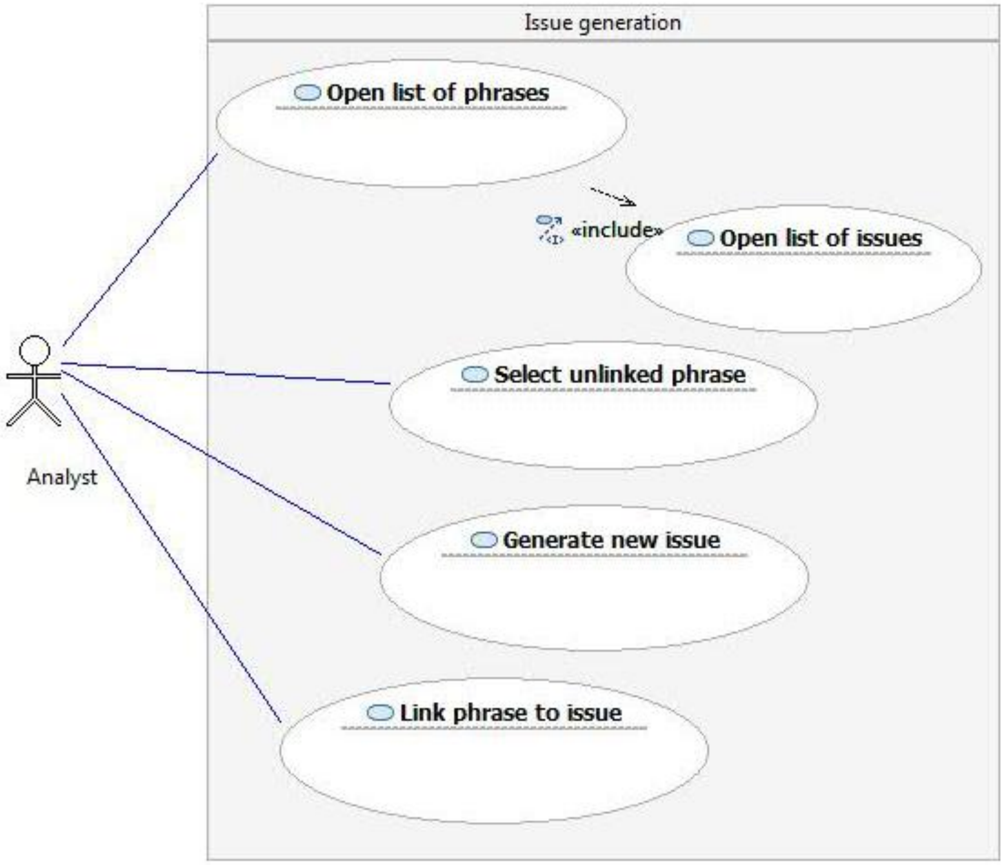


Figure 13 Use case "Issue Generation"

Use Case Diagram Name	Issue generation
Related Requirements	T-39
Goal in Context	The tool allows for selection of phrases from the list of phrases and for enabling to either link phrases to existing issues or to create new issues based on at least one phrase.
Preconditions	1. The Analyst must be registered at the system. 2. The Analyst must be logged in. 3. The analyst receives the view of the data according to the rights he/she owns. 4. The analyst can work on the data according to the rights he/she owns.
Successful End Condition	The CAQDAS is viewed and able to generate new issues from existing phrases and to link new phrases to existing issues.
Failed End Condition	The CAQDAS cannot be viewed. The CAQDAS can be viewed but is not able to generate new issues from existing phrases. The CAQDAS can be viewed and is able to generate new issues from existing phrases but not to link new phrases to existing issues.
Primary Actors	Analyst
Trigger	The Analyst found a phrase, which is not similar to any of the existing issues in the list; hence he/she generates a new issue based on the phrase. The Analyst found a phrase, which is similar to one of the existing issues in the list; hence he/she links the new phrase to the existing issue.
Main Flow	**Step. Action**
	1. The Analyst opens a list of phrases. 1.1. *Include: Open list of issues.* The Analyst opens the list of issues. 2. The Analyst selects one phrase, which is not yet linked to an issue. 3. The Analyst generates a new issue based on the phrase. 4. The Analyst links the selected phrase to an existing issue.

Table 26 Description of the “Issue Generation” use case

5.6.1.3. Generation of relations and relation clusters

Coding a text passage means to define the phrase by assigning a keyword to it and fixing its position within the text. Different phrases (text passages) may concern the same matter, and, therefore, are grouped into issues. Each issue represents a cluster of phrases of the same matter, i.e. it does not only correspond to one specific phrase but the number of phrases referring to the same or to a very similar matter. Issues represent a set of coded text passages (i.e. phrases) extracted from the investigated texts. Accordingly, the list of issues is related to the list of phrases. Each issue consists of a number of phrases (1 to n). The issue is linked via the code of each phrase to the corresponding phrases. As the code of the phrases determines the text from which the phrase was extracted originally and the position within the text, coding and clustering generates traceability. Traceability is very important to avoid de-contextualisation. Each document comes up with new insights regarding the characteristics of the issue. The issue becomes clearer with each new similar or contrary phrase discovered. If necessary an issue is separated into several issues depending on the discoveries made based on the phrases assigned to the issue. Overall, merging and comparison of phrases advance the definition of the issue.

Relations among issues can be identified either by a text passage (i.e. phrase that is identified, extracted and coded by the analyst) or by the expertise of the analyst.

Relations can be detected only as relations between phrases of two different issues. Several relations among issues are possible. All relations identified between two different issues are clustered to relation clusters according to their similar meanings. If at least two phrases of two different issues are related with each other then at least one relation cluster exists. The relation between issues at the issue level is therefore called relation cluster. Since a text-based relation always describes the relation between two phrases of different issues, relation clusters between issues can be queried via the issue itself as the phrases are included (i.e. linked via unique identifier to the issue) in the issue.

The use cases that refer to generation of relations and relation clusters are present in Figure 14.

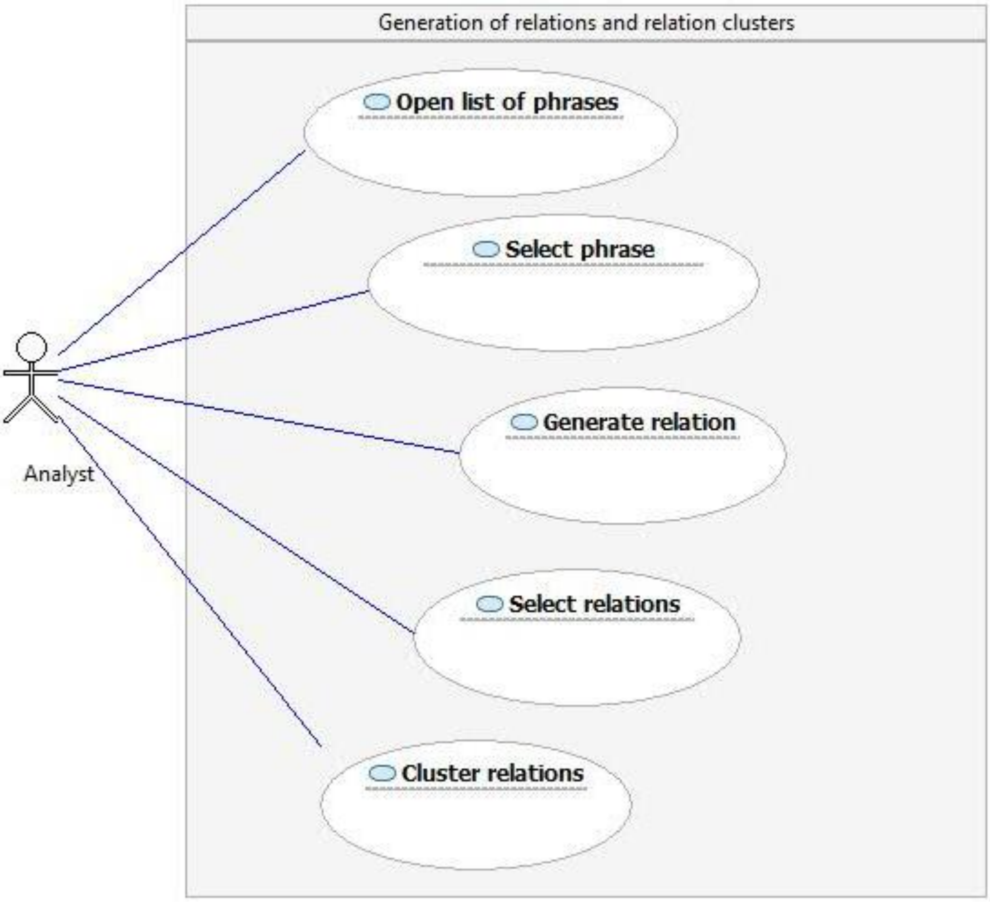


Figure 14 Use case “Generation of Relations and Relation Clusters”

Use Case Diagram Name	Generation of relations and relation clusters
Related Requirements	-
Goal in Context	The tool allows for selection of phrases from the list of phrases thereby defining if the phrase is a relation among two other phrases. Similar relations among the same two issues are clustered to relation clusters.
Preconditions	1. The Analyst must be registered at the system. 2. The Analyst must be logged in. 3. The analyst receives the view of the data according to the rights he/she owns. 4. The analyst can work on the data according to the rights he/she owns.
Successful End Condition	The CAQDAS is viewed and able to generate relations from existing phrases and to cluster similar relations among the same two issues to relation clusters.
Failed End Condition	The CAQDAS cannot be viewed. The CAQDAS can be viewed but is not able to generate relations from existing phrases. The CAQDAS can be viewed and is able to generate relations from existing phrases but not to cluster similar relations among issues to relation clusters.
Primary Actors	Analyst
Trigger	The Analyst found a phrase, which represents a relation among two other phrases; hence he/she generates a new relation based on the phrase. The Analyst found several similar relations among the same two issues; hence he/she clusters these similar relations among the same two issues to relation clusters.
Main Flow	**Step. Action**
	1. The Analyst opens list of phrases. 2. The Analyst selects one phrase. 3. The Analyst generates a new relation based on the phrase. 4. The Analyst selects relations between phrases (i.e. similar phrases among two issues). 5. The Analyst clusters the selected relations to a relation cluster.

Table 27 Description of the "Generation of Relations and Relation Clusters" use case

5.6.1.4. Expertise-based relations

Up to this step, the qualitative data analysis and the results extracted are easily traceable and replicable as they are extracted from text, i.e. explicit knowledge. Further elaboration results in steady decrease of traceability and replicability, because the subsequent interpretation of data bases mainly on the knowledge of the experts, who are structuring, synthesizing and interpreting the material. As a consequence, the management of issues and their interrelations became more complex. Figure 15 presents the use case diagram with expert users inserting relations between issues based on their intrinsic knowledge.

Expertise-based relations are relations not explicitly mentioned in the documents but are based on the expertise of the analyst. For the expertise of the specialist, which is intrinsic knowledge, no reference can be given except an identifier to classify the analyst from whom the expertise comes from. To enhance the quality of qualitative data analysis, relations have to be considered that are extracted from text (i.e. scenarios) as well as those grounding on implicit expert knowledge. To make this difference visible the distinction of both cases should be explicitly documented. This means, that the origin of the relation (i.e. either implicit expert or explicit text knowledge) has to be stored as well to make this distinction visible. Work on the content (i.e. metadata) needs to be done manually.

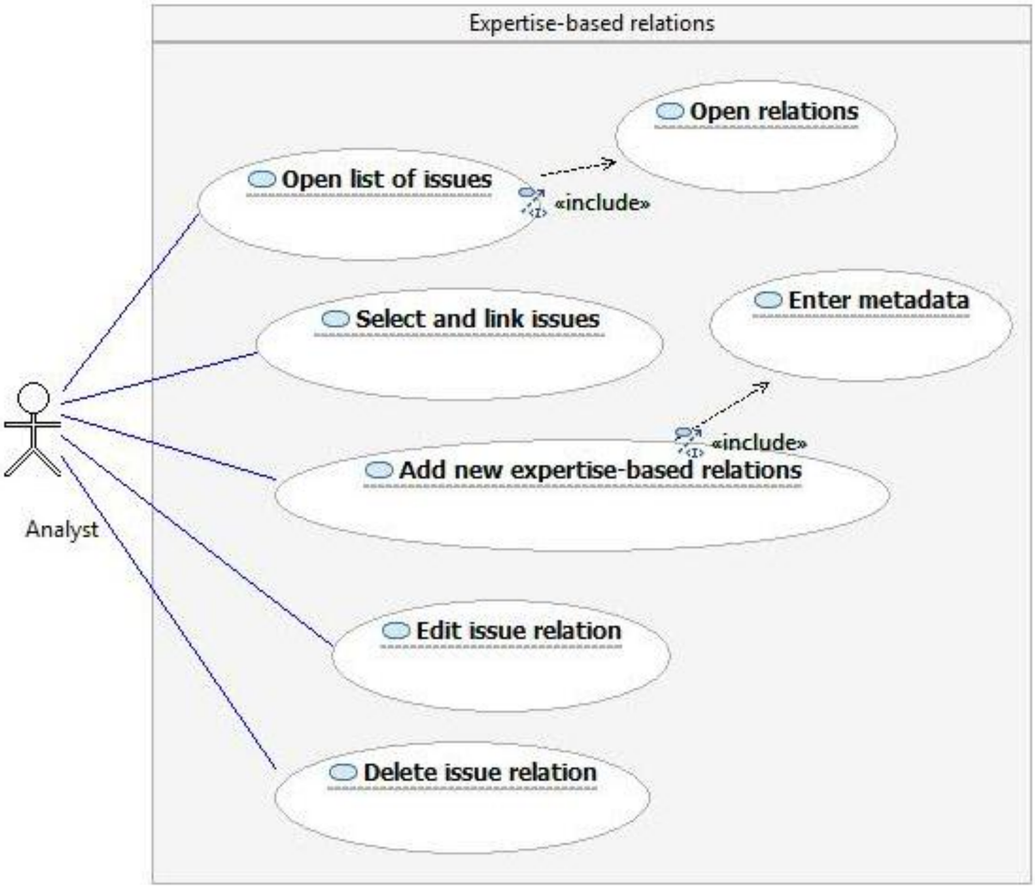


Figure 15 Use case “Inserting Expertise-based Relations”

Use Case Diagram Name	Inserting expertise-based relations.
Related Requirements	-
Goal in Context	The CAQDAS allows inserting relations between issues, which are not derived from natural language description but from the expertise of the analyst.

Preconditions	1. The analyst must be registered at the system 2. The analyst must be logged in 3. The analyst receives only the view of the data according to the rights he/she owns 4. The analyst can only work on the data according to the rights he/she owns
Successful End Condition	The CAQDAS is viewed and can be used.
Failed End Condition	The CAQDAS does not allow inserting relations between issues based on the expertise of the Analyst
Primary Actors	Analyst
Trigger	The Analyst recognised the existence of relations between issues which are not found in any natural text description, but which he/she knows from his/her expertise.
Main Flow	**Step. Action**
	1. The Analyst opens a list of issues, including: 1.1. *Include: Open relations.* The Analyst can view expertise-based relations. 2. The Analyst selects two issues (i.e. those issues between the relation exists) and links them with the relation data entry to be created. 3. The Analyst adds new expertise-based relations 3.1. *Include: Enter metadata.* The Analyst can enter metadata related to relations. 4. The Analyst can edit relations between issues. 5. The Analyst can delete issue relation.

Table 28 Description of the “Inserting Expertise-based Relations” use case

5.6.2. Quantitative Analysis of Documents

The scenario analysis is supported by quantitative data analysis (i.e. statistics). During this process the traceability of data is assured. The aim of quantitative analysis of documents is to advance the model enriching it by facts and inferences about the subject under study. Related use cases are presented in Figure 16.

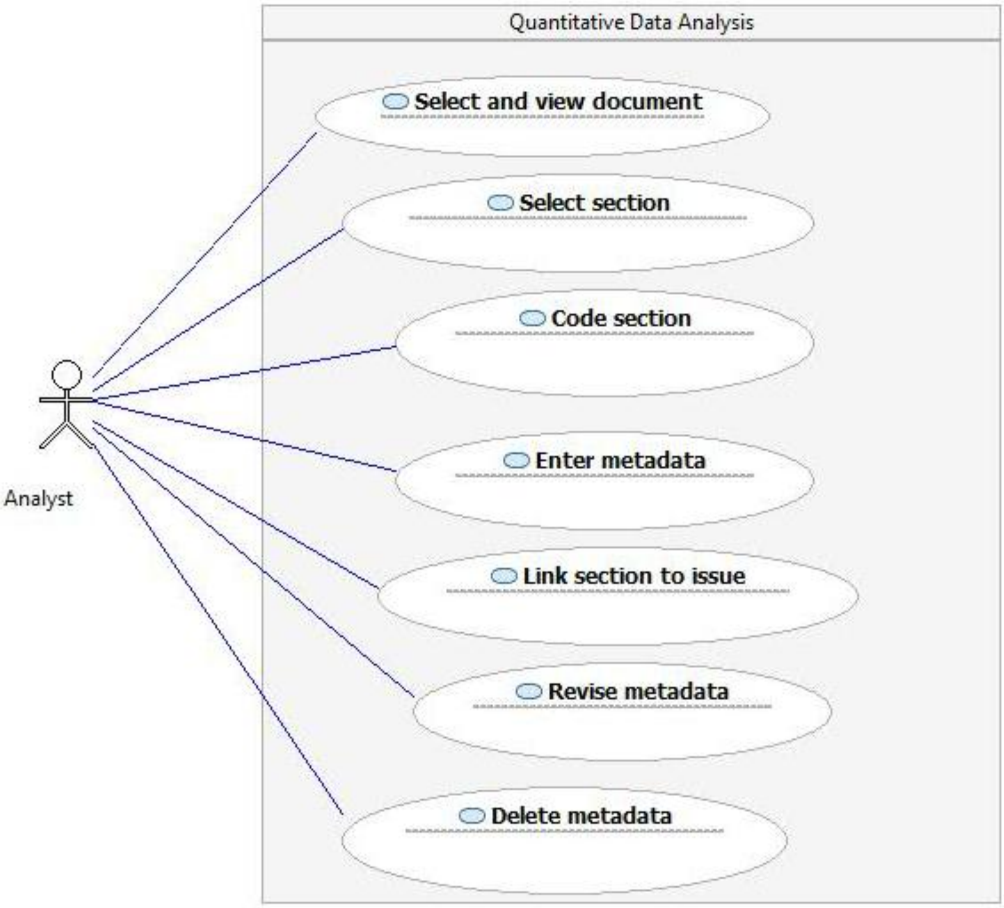


Figure 16 Use case "Quantitative Data Analysis"

Use Case Diagram Name	Quantitative Data Analysis of documents
Related Requirements	-
Goal in Context	Quantitative Data Analysis of documents (i.e. statistics) is to improve the model through facts and to draw inferences about the process or population being studied. The Quantitative Data Analysis provides tools for prediction and forecasting using data and statistical models (i.e. dealing with uncertainties through showing probabilities).
Preconditions	1. The analyst must be registered at the system. 2. The analyst must be logged in. 3. The analyst receives the view of the data according to the rights he/she owns. 4. The analyst can work on the data according to the rights he/she owns.
Successful End Condition	The document analysis tool is viewed and facilitates the analysis.
Failed End Condition	The Quantitative Data Analysis tool does not render the required features/services, i.e. it delivers useless input for the CCD and simulation models.

Primary Actors	Analyst
Trigger	The first background documents for the policy case are available, which are including quantitative data, so that the analyst can initiate the quantitative data analysis.
Main Flow	**Step. Action**
	1. The Analyst selects and views the background document. 2. The Analyst goes through the document and highlights relevant sections. 3. The Analyst codes the sections. 4. The Analyst enters metadata of analyzed section. 5. The Analyst can link the section (quantitative information) to an existing issue. 6. The Analyst can revise metadata of analyzed documents. 7. The Analyst can delete metadata.

Table 29 Description of the "Quantitative Data Analysis" use case

5.6.3. Network Visualisation

Visualisation of dependencies shall be developed in order to visualize and distinguish between the different types of results of scenario analysis and their relations or implications. The visualisation can be either a network or a table. All use cases referring network visualization are depicted in Figure 17.

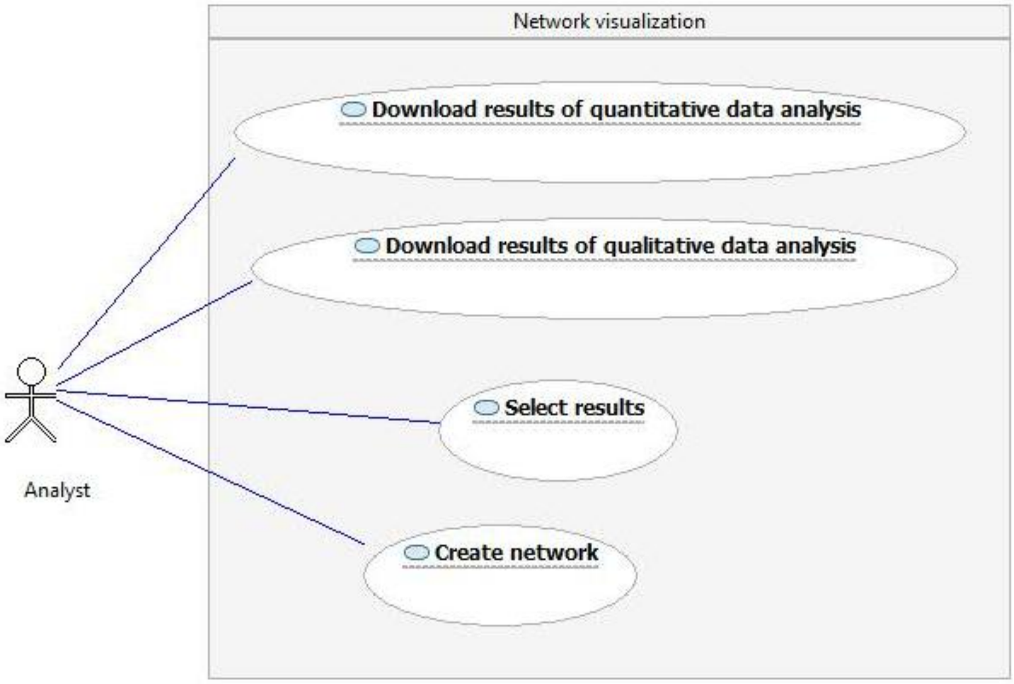

Network visualization
Download results of quantitative data analysis
Download results of qualitative data analysis
Select results
Create network
Analyst

Figure 17 Use case "Network Visualisation"

Use Case Diagram Name	Network visualisation.
Related Requirements	-
Goal in Context	The visualisation allows for presenting dependencies between different kinds of results.
Preconditions	1. The analyst must be registered at the system 2. The analyst must be logged in 3. The analyst receives the view of the data according to the rights he/she owns 4. The analyst can work on the data according to the rights he/she owns
Successful End Condition	The visualisation tool is viewed and can be used to visualise network(s).
Failed End Condition	The visualisation tool is not rendered, the dependencies cannot be inserted.
Primary Actors	Analyst
Trigger	The Analyst recognised the existence of dependencies between results.
Main Flow	**Step. Action**
	1. The Analyst downloads quantitative data analysis. 2. The Analyst downloads qualitative data analysis. 3. The Analyst selects results. 4. The Analyst creates network.

Table 30 Description of the “Network Visualisation” use case

5.7. POLICY MODELLING

At the start, the policy modeller needs to identify his/her goal. Later the policy modeller has to extract stakeholders with their descriptions and develop the environmental rules and facts for the model. The facts are elements that carry information, which is founded in a source. For example, actors as depicted in the “social network graph” are transformed into the policy model by describing those using so called fact templates. The instantiation of an actor in a policy model (program code) is called a fact. As not all facts relevant to the policy model can be explicitly found in text, such facts are known as *magic facts*. These are based on the expert’s intrinsic knowledge and do not have an explanatory statement in the base texts.

In developing rules or inspecting rules, it is useful to be able to select one or more clauses and then either fetch them (if all clauses are on the database) or retrieve them (if there are some clauses involving calculation such as > or <). Useful if a rule that was expected to fire did not or if a new rule is being implemented at a paused time step during a simulation.

The data dependency graph, as another important element of the policy model, has been described. The data dependency graph consists of rules and facts, i.e. the rule-dependency graph is extended with the data elements (the facts).

At the end the rule dependency graph needs to be check for consistency and this should be done automatically by appropriate software.

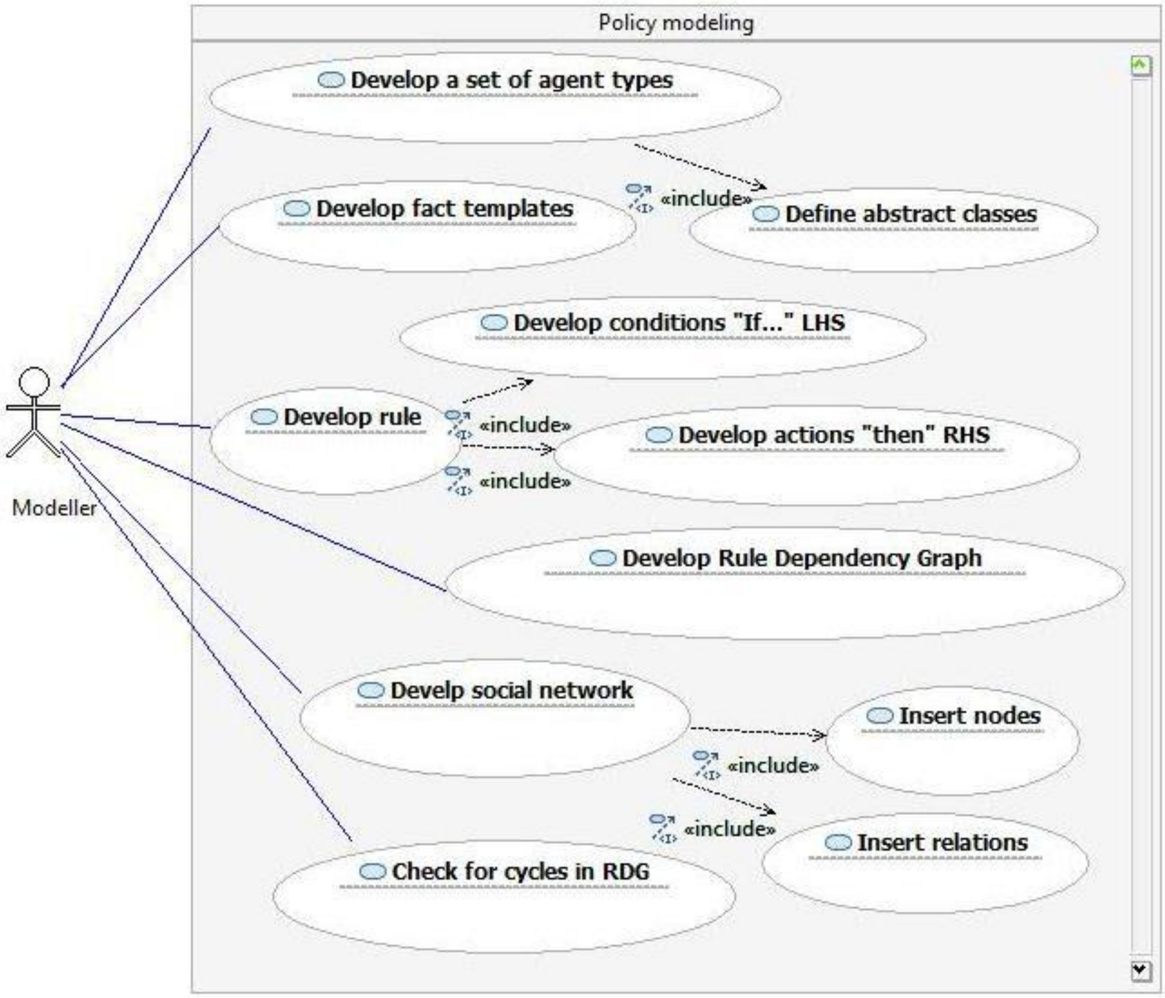


Figure 18 Use case “Policy Modelling”

Use Case Diagram Name	Policy modelling
Related Requirements	I-14, I-20, I-26, I-30, I-39, I-40; FR01_PM, FR02_PM, FR03_PM, FR04_PM, FR05_PM, FR06_PM, FR07_PM, FR08_PM, FR09_PM, FR10_PM, TP-1
Goal in Context	Functionality allowing the development of the model (i.e. agent types, fact templates, Rule Dependency Graph, social networks) is provided.
Preconditions	1. The Modeller must be registered at the system. 2. The Modeller must be logged in. 3. The Modeller receives the view of the data according to the rights

	he/she owns. 4. The Modeller can work on the data according to the rights he/she owns.
Successful End Condition	The policy modelling tool is provided.
Failed End Condition	The policy modelling tool is not rendered.
Primary Actors	Modeller
Trigger	The Modeller initiates policy modelling.
Main Flow	**Step. Action**
	1. The Modeller develops a set of agent types; 1.1. *Include: Define abstract classes.* The Modeller defines abstract classes of agents. 2. The Modeller develops fact templates used in the model. 3. The Modeller develops rule. 3.1. *Include: Develop conditions “If...” LHS.* The Modeller develops conditions “If..”, 3.2. *Include: Develop* actions “then” *RHS.* The Modeller develops actions “then” used in the model. 4. *Develop Rule/Data Dependency Graph.* The Modeller creates rules dependency graph of rules and data used in the model. 5. The Modeller creates social network of agents; 5.1. *Include: Insert nodes.* The Modeller inserts nodes of the social network 5.2. *Include: Insert relations.* The Modeller inserts relations between nodes. 6. The Modeller checks for cycles in RDG.

Table 31 Description of the “Policy Modelling” use case

5.8. SIMULATION

In the next step the modeller starts the simulation. At first it is essential to have a possibility to set up initial rules, facts and parameters of simulation. After stopping the simulation the Modeller can restart (to continue simulation from the last stop point) it or start again (to start simulation from beginning). There is the practical difference between restarting and starting a simulation. While restarting, it is possible to establish rules and experiment with them on the fly. After the simulation is stopped by modeller manually or by reaching the end state, it is possible to preview all results, search for specific events, and revise the settings. Starting the simulation in experimental run generates the output, which will be used in evaluation of simulation.

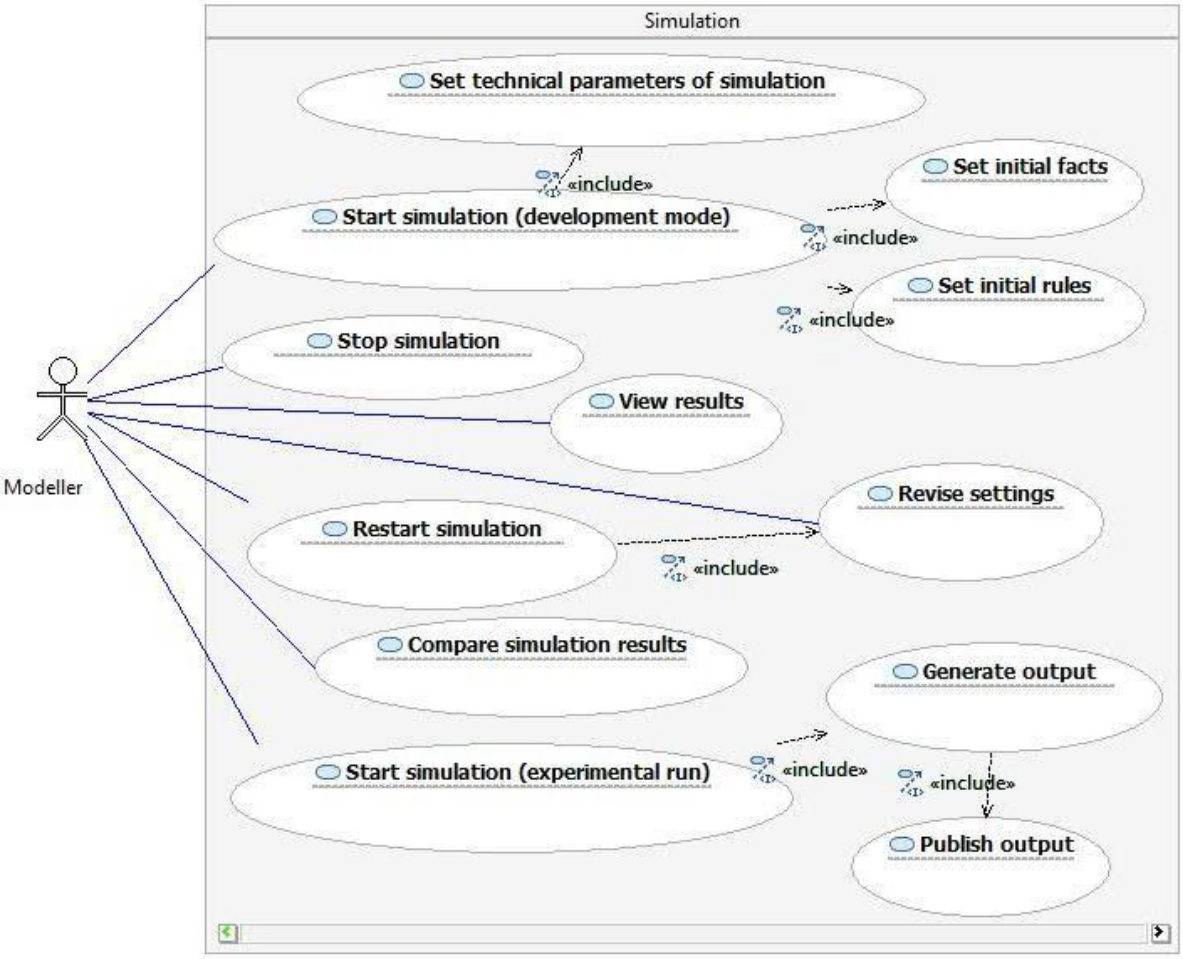


Figure 19 Use case "Simulation"

Use Case Description Name	Simulation
Related Requirements	I-18, I-20, I-24, I-27, T-16, T-19,T-20, T-22, T-23, FR11_PM, FR12_PM, FR13_PM, FR14_PM, FR15_PM, FR16_PM, FR17_PM, FR18_PM, FR19_PM, FR20_PM, FR21_PM, FR25_PM, FR26_PM, TP-3, TP-5,
Goal in Context	The tool enables to run simulations including possibility to set technical parameters, initial rules and facts.
Preconditions	1. The Modeller must be registered at the system. 2. The Modeller must be logged in. 3. The policy model is available. 3. The Modeller receives the view of the data according to the rights he/she owns. 4. The Modeller can work on the data according to the rights he/she owns.
Successful End Condition	The simulation tool is viewed and can be used (to perform simulations).
Failed End Condition	The simulation tool is not rendered and the simulation cannot be performed.
Primary Actors	Modeller

Trigger	The Modeller initiates simulation.
Main Flow	**Step. Action**
	1. The Modeller starts development of simulation in development mode, including: 1.1. *Include: Set technical parameters of simulation.* The Modeller defines initial parameters that are going to be simulated. 1.2. *Include: Set initial facts.* The policy Modeller defines the initial facts of the model. 1.3. *Include: Set initial rules.* The policy Modeller defines the initial rules. 2. The Modeller can restart simulation. 2.1. *Include: Revise settings:* The Modeller uses different configurations. 3. The Modeller can revise settings. 4. The Modeller stops simulation. 5. The Modeller can view results of simulation. 6. The Modeller can compare results derived from different simulations. 7. The Modeller starts simulation in experimental run. 7.1. *Include: Generate output.* The Modeller generates output of simulation, including: 7.1.1.*Include: Publish output.*

Table 32 Description of the “Simulation” use case

5.9. EVALUATION

For validation/evaluation, end users (e.g. stakeholders) need to access information generated by the simulation model. Validation aims at checking the consistency and, precision of both evidence-based user generated scenarios and simulation results. End users can comment and evaluate presented simulation results. In case of any uncertainties end user can ask modeller.

During the evaluation users are able to play role-playing games using specifically designed user interface, which will allow them to change all necessary aspects and parameters, together with taking their decisions during the simulation steps. Playing games within policy modelling tool can be performed in order to acquire knowledge about the system and learning how the simulation works. The educative games will base on the simulations in which users will be allowed to change parameters of the simulation and observe how the magnitudes of change influence the development of the future states. The game should be interactive so the user is asked to make the decision every few time steps of the simulation run.

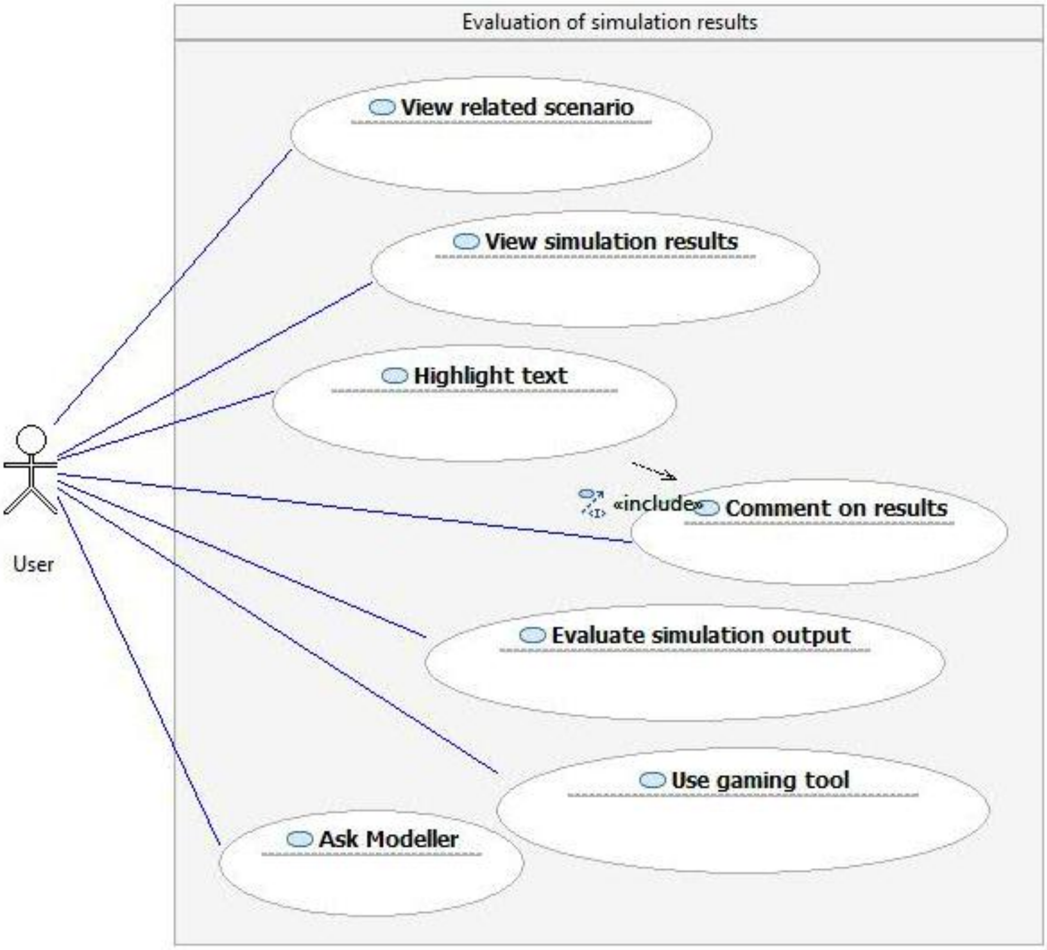


Figure 20 Use case "Evaluation of Simulation Results"

Use Case Diagram Name	Evaluation of simulation results.
Related Requirements	I-17, I-20, I-29, FR22_PM, FR23_PM, FR24_PM, FR27_PM, T-25, T-32, T-33
Goal in Context	The tool enables the User to evaluate the simulation results, to use gaming tool and to ask Modeller about details.
Preconditions	1. The User must be registered at the system. 2. The User must be logged in. 1. The output of experimental run of simulation is available. 2. The User receives only the view of the data according to the rights he/she owns. 3. 4. The Modeller can only work on the data according to the rights he/she owns.
Successful End Condition	The tool is viewed and can be used.

Failed End Condition	The tool is not rendered and the User cannot see the model-based scenario.
Primary Actors	User
Trigger	The User initiates evaluation process.
Main Flow	**Step. Action**
	1. The User views simulation results. 2. The User can view the user generated scenario related to the simulation outcome. 3. The user can highlight part of the text. *Include: Comment on results.* The User can express his/her opinion. 4. The User can comment on the results of simulation. 5. The User can evaluate the simulation output. 6. The User can take the role of agent in the simulation (gaming mode) and see how his/her behaviour influences the outcome of the simulation. 7. The User can ask Modeller questions regarding the outcome of the simulation.

Table 33 Description of the "Evaluation of Simulation Results" use case

5.10. NEW USER REQUIREMENTS BASED ON USE CASE ANALYSIS

Requirement ID: UC-1 **Requirement Type:** Functional **Priority:** Must-have
Name: Rights management
Description: The Initiator/facilitator can assign rights to users and modify them during the run of the project. The initiator can assign other users the right to assign rights as well.
Measurement indicators: Available functionality.

Requirement ID: UC-2 **Requirement Type:** Functional **Priority:** Should-have
Name: Invitation – send and receive
Description: The authorised user can send an invitation to registered as well as unregistered user. The invited User gets e-mail with a link which he/she should click if he/she wants to accept the invitation.
Measurement indicators: Available functionality.

Requirement ID: UC-3 **Requirement Type:** Functional **Priority:** Nice-to-have
Name: Send request for invitation
Description: Uninvited User who wants to take part in the project sends a request to the facilitator with the description who he/she is and why he/she is interested in taking part in the project. The request is in form of the template.
Measurement indicators: Available functionality.

Requirement ID: UC-4 **Requirement Type:** Functional **Priority:** Must-have
Name: Initiate project
Description: The facilitator is able to initiate a project. He/she is able to generate an initial description of the project (name, abbreviation, and outline of the project) and upload relevant documents.
Measurement indicators: Available functionality.

Requirement ID: UC-5 **Requirement Type:** Functional **Priority:** Must-have
Name: Update initial description of the project
Description: The Facilitator and users can update initial description of the project in order to extend/explain communicated information.
Measurement indicators: Available functionality.

Requirement ID: UC-6 **Requirement Type:** Functional **Priority:** Should-have
Name: Generation of relation
Description: The CAQDAS tool should provide the possibility to generate relations and relation clusters in a manual way. When modeller finds in texts the relation between phrase A and B, where phrase A and B belong to two different issues (i.e. phrase A belongs to issue A and the other phrase B belongs to issue B) he/she selects phrases and generates relation. The phrase that describes the relation between phrase A and B is also coded and linked to phrase A and B. The analyst can also select relations between phrases and cluster relations.
Measurement indicators: Available functionality.

Requirement ID: UC-7 **Requirement Type:** Functional **Priority:** Should-have
Name: Expertise-based relation
Description: The CAQDAS tool allows inserting relations between issues, which are not derived from natural language description but from the expertise of the Analyst. The Analyst opens the list of issues and relations. Based on his/her expertise the Analyst selects two issues (i.e. those issues between the relation exists), links them and adds new expertise-based relations. The Analyst can enter metadata related to relations. At any time the Analyst can edit and delete relations between issues.
Measurement indicators: Available functionality.

Requirement ID: UC-8 **Requirement Type:** Functional **Priority:** Should-have
Name: Quantitative data analysis
Description: The Quantitative Data Analysis provides tools for prediction and forecasting using data and statistical models (i.e. dealing with uncertainties through showing probabilities). The Analyst goes through the document and highlights relevant sections, codes them and links the section (quantitative information) to an existing issue. The Analyst can enter metadata of analysed section, revise it as well as delete.
Measurement indicators: Available functionality.

Requirement ID: UC-9 **Requirement Type:** Functional **Priority:** Should-have
Name: Network visualisation
Description: The visualisation of network allows for presenting dependencies between different kinds of results.
Measurement indicators: Available functionality.

Requirement ID: UC-10 **Requirement Type:** Functional **Priority:** Should-have
Name: Development of social network
Description: The development of social network requires extraction of actors, their relations, interests, constrains, etc.
Measurement indicators: Available functionality.

6. ARCHITECTURAL VIEWS AND PERSPECTIVES

In order to present designed architecture, a “divide and conquer” approach is employed – the overall architecture is described by a set of architectural views (each one representing a particular aspect of the architecture) and perspectives (each one representing a particular quality property which is orthogonal to the used views).

6.1. FUNCTIONAL VIEW

6.1.1. Design considerations

Client-server vs. service oriented architecture

Client and server are software entities which are in a close relationship. In any exchange relation, the client initiates a request and the server responds adequately – it interprets a communicated request and then attempts to fulfil it. Usually, a server can serve several clients. On the other hand, service oriented architecture emphasizes design of architecture components as modular services which can be searched for, discovered and utilised. Selected services communicate using a standard communication scheme (e.g. SOAP-based or REST-based) to exchange messages. Following this principle, services can be composed to form higher-level services. As a result, this approach enables to produce highly flexible architecture types which are ready for distributed deployment. Since the project has identified a relatively stable understanding how required system functionality should be used to support users, the high flexibility level provided by the service oriented architecture is not necessary. To reduce overall complexity and effort connected with it, the project partners have opted for older and simpler client-server architecture, which is fully sufficient for the project.

Two tiers vs. three tiers

Two tier architecture splits all the architecture into two separated layers. The most often, presentation capabilities and application business logic are packed together into one tier while the other tier is dedicated to managing and processing all data the application deals with. The most compelling advantage of this architecture type is its simplicity and application development speed. On the other hand, it works well in relatively homogeneous environments only. Although more laborious, three tiers enable additional separation – the most often, presentation capabilities are separated from business logic. The middle tier can play a role of an intermediary centralising some functionality and providing it for different parts of the client part of the architecture. Since the project tries to reuse different existing tools to meet as many user requirements by them as possible, it is expected that the selected tools will form a heterogeneous environment and therefore the three tier architecture has been selected. In addition, finer granularity enables working on different system parts in parallel more easily.

Coupling vs. cohesion

In order to assess the quality of software architectures, different metrics have been designed, measuring a wide range of parameters from coverage of functional requirements to extensibility aspects. This includes the attributes coupling and cohesion as well. Coupling is defined as the degree of connections between different system components (inter-component links), while cohesion is the degree of relationship between the constituents of a system component (intra-component links). A conventional wisdom is to minimise coupling and maximise cohesion – to reduce interdependencies between system components to bare essentials in order to keep coupling at the minimum level for maintainable and secure systems (loose coupling) and pay for it with high cohesion. Recently, [Booch et al., 2007] suggests an alternative - a “functional cohesion”, meaning that all the constituents work together to fulfil one specific task. The OCOPOMO system architecture was indeed designed with a “functional cohesion” in mind, making the components as functionally self-contained as possible with a very loose coupling between them.

Degree of centralisation

A special care has been taken to design an architecture in a way that minimizes the degree of centralization inherent. There is not a single point of control defined in the prospective OCOPOMO system. The logic of the presented design is a control structure spread across all components, leading to an implicit (local) control structure(s). System components employ a flexible communication model – if a software manager needs a service then it utilises a manager providing the requested service. No centralised component deciding who should communicate with whom and when is incorporated in the architecture.

6.1.2. Overall functional architecture

An overview of the overall architecture is given in Figure 21. The following subsections of this document deal with a detailed description of all defined parts.

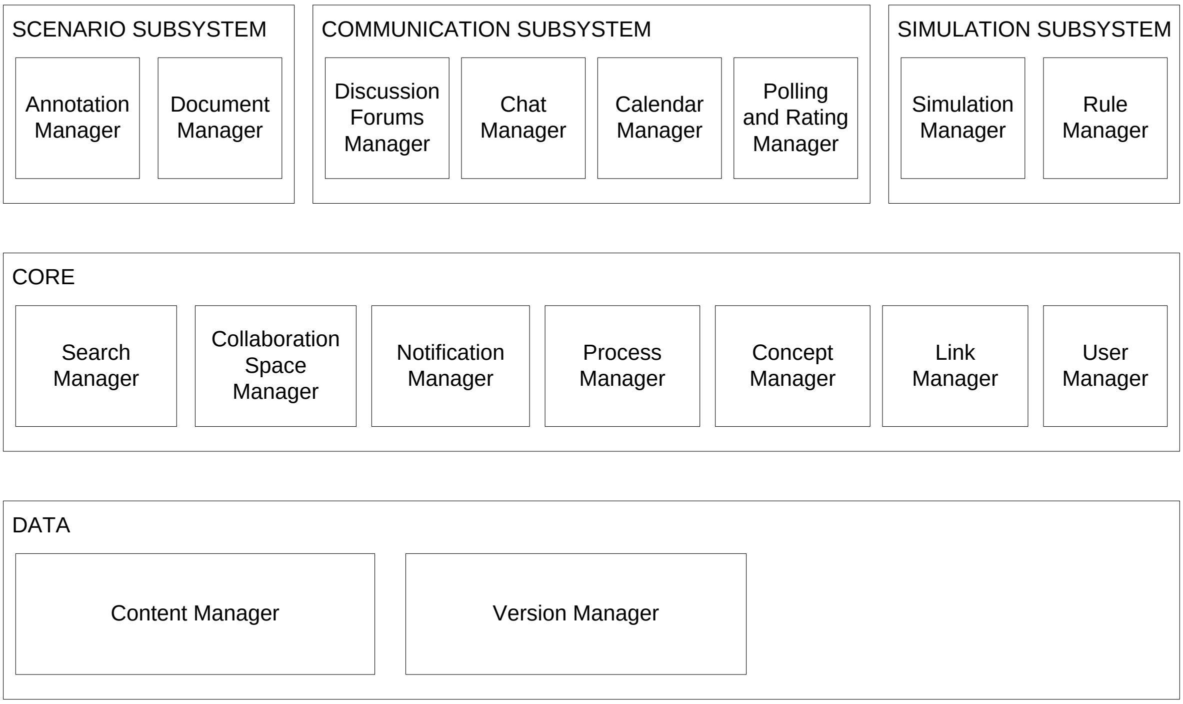


Figure 21 Overall architecture of the OCOPOMO platform

The OCOPOMO system architecture consists of three main layers:

- TOOLS - layer of components which are responsible for work of particular tools within the system and its functional user interfaces. This part can be also structured into three parts:

    - *Communication subsystem* - functionality that covers communication, collaboration and cooperation-based features of the platform. These tools are able to support also other subsystems with their features (as they are not directly related to policy modelling process in OCOPOMO).
    - *Scenario subsystem* - functionality that fulfils scenario generation and analysis part of the policy modelling process in OCOPOMO (other functionality from communication subsystem could be also used, but essential are those for scenario creation and analysis).
    - *Simulation subsystem* - functionality which is important for modellers in order to create, update, visualize and execute simulation models (agents, rules, etc.) within the OCOPOMO platform.
- CORE - subsystem called the OPOCOMO Core is dedicated to processing all the data in the system and supporting the tools layer with any business/data logic related to project resources, metadata and processes, as well as to support functionality (business logic and user interface) which has wider scope as an individual tool and more aspects should be combined there (e.g. federated search, system wide notification, process/space initiation, user profile management, etc.).
- DATA - data-level of architecture which is responsible for managing of storage and sharing of particular content and its versioning.

The presented layers can be mapped onto a standard three tier structure in the following way: CORE managers together with subsystems' managers correspond to the middle tier (usually called business logic tier), user interfaces of subsystems' and CORE managers correspond to the upper tier (usually called presentation tier), and DATA managers correspond to data tier.

Discussion Forums Manager

Short description	Responsible for providing discussion forum functionality (through a collaboration space) within the OCOPOMO system.
Expected functionality	Creation, editing and deleting of discussion forums, threads (topics) and messages with rating/tag and notification functionality reused.
Input/output components	Collaboration Space Manager, Search Manager, Notification Manager, User Manager, Polling and Rating Manager, Content Manager, Chat Manager.

Table 34 Discussion Forums Manager

This part of the system is responsible for off-line communication based on the discussion forums functionality, which can be used within the collaboration space of the OCOPOMO system. Basic functionality includes possibility to create new forums, threads (topics) and messages, as well as edit and delete them. Discussion forums are part of the collaboration space and specific process, therefore important connection is to *Collaboration Space Manager*. Search within discussion forums is supported by the *Search Manager*. Rating functionality is reused for discussion evaluation (relevance feedback) and analysis, where *Polling and Rating Manager* is helpful. In case of need for notification support with different channels (e.g. email notification, RSS, news, newsletter) *Notification Manager* is reused. *User Manager* provides particular access rights for any actions within the manager. For storage and retrieval of manager-specific data *Content Manager* is used for content repository functions. *Chat Manager* is able to reuse discussion forums functionality to create off-line discussion after finishing chat session.

Chat Manager

Short description	Responsible for providing chat functionality (through a collaboration space) within the OCOPOMO system.
Expected functionality	Creation of (context-specific) chats, save history of chat as a document, adding messages to chat, create discussion forum related to chat (on demand).
Input/output components	Collaboration Space Manager, Document Manager, Discussion Forums Manager, Search Manager, User Manager.

Table 35 Chat Manager

This part of the system is responsible for on-line communication based on the chat functionality, which can be used within the collaboration space of the OCOPOMO platform. Basic functionality includes possibility to create new chat (in a specific context) and add messages into chat. *Collaboration Space Manager* provides standard connection to shared collaboration space and process-specific data. *Document Manager* is used for saving of chat's history as a document. Search within chat is supported by the *Search Manager*. Users are also able to create discussion forums related to the chat, where communication can continue in off-line mode using *Discussion Forums Manager*. *User Manager* provides particular access rights for any actions within the manager.

Calendar Manager

Short description	Responsible for providing shared calendar functionality (through a collaboration space) within the OCOPOMO system.
Expected functionality	Adding, editing and removing events (which are process or context specific) in shared calendar, reminder functionality (reused notification).
Input/output components	Collaboration Space Manager, Notification Manager, User Manager, Search Manager, Content Manager.

Table 36 Calendar Manager

This manager is responsible for coordination-like functionality – shared calendar and agenda, which can be used within the collaboration space of the OCOPOMO system. Basic functionality includes adding, editing and removing shared calendar events (process or context specific). Calendar is a part of the collaboration space and specific process, therefore important connection is to *Collaboration Space Manager*. Search within calendar events is supported by the *Search Manager*. In case of need for notification support (e.g. reminder functionality) *Notification Manager* is reused. *User Manager* provides particular access rights for any actions within the manager. For storage and retrieval of manager-specific data *Content Manager* is used for content repository functions.

Polling and Rating Manager

Short description	Responsible for providing polling and (in more general way) rating functionality (through a collaboration space) within the OCOPOMO system.
Expected functionality	Creation and managing of polls, rating API and user interfaces for different types of objects available for other components (e.g. discussion forums).
Input/output components	Collaboration Space Manager, Discussion Forums Manager, Content Manager, Notification Manager, User Manager.

Table 37 Polling and Rating Manager

This part of the system is responsible for the creation of polls and rating functionality, which can be reused within the system. Basic functionality includes creation and managing of polls as a part of the collaboration space and specific process (connection to *Collaboration Space Manager*). Rating functionality, also managed by this component, can be reused (as API and user interface fragments) for different objects within the system, especially for *Discussion Forums Manager*. *User Manager* provides particular access rights for any actions within the manager. For storage and retrieval of manager-specific data *Content Manager* is used for content repository functions. Notification functionality, provided by *Notification Manager*, is used for announcing of polls and their results.

Document Manager

Short description	Responsible for providing content management functionality (through a collaboration space) within the OCOPOMO system, mostly for scenario creation phase of the process and linking of conceptual models with information from real documents.
Expected functionality	Creation, opening, deleting and tagging of documents, inserting different resources into the system (as documents of different formats and/or links to resources), versioning of documents.
Input/output components	Annotation Manager, Collaboration Space Manager, User Manager, Search Manager, Content Manager, Version Manager, Chat Manager, Notification Manager, Simulation Manager.

Table 38 Document Manager

This manager is responsible for providing document management for users, which can be used within the collaboration space of the OCOPOMO system, for any purposes of sharing resources and information (data). Basic functionality includes creation, opening, deleting and tagging documents, input of different resources into the system, all as a part of collaboration space (*Collaboration Space Manager*). One of the main functions is also support of scenario generation and analysis, therefore *Annotation Manager* is directly connected to this manager. Search within documents and resources are supported by the *Search Manager*. *User Manager* provides particular access rights for any actions within the manager. For storage and retrieval of manager-specific data *Content Manager* is used for content repository functions. *Version Manager* is used for support of versioning functionality. In case of need for notification support (e.g. reminder functionality) *Notification Manager* is reused. Other components are also able to reuse document management (if needed), especially if they are connected through collaboration space or policy modelling process.

Annotation Manager

Short description	Annotation tool - transformation from unstructured text (scenarios) to structured information (CCD - Consistent Conceptual Description), important for policy modellers in next steps of the process.
Expected functionality	Open scenario (document), scenario analysis using annotation tool, highlighting the text and creation of specific objects (annotations), creation of relations about/between the objects and their groups.
Input/output components	Document Manager, Concept Manager, Link Manager, Process Manager, User Manager, Search Manager, Content Manager.

Table 39 Annotation Manager

This manager is responsible for one part of the process, where scenario is analysed and structured information (suitable for modellers) is provided in a specific conceptual format. Basic functionality includes opening the scenario, analysis using annotation tool, creation of specific objects (annotations), creation of relations between objects and their groups, which are then a part of the conceptual description about the current understanding of problem area. Search within the document (scenario), annotations, relations and other relevant objects are supported by the *Search Manager*. Relevant objects are reused and managed also within *Concept Manager* and *Link Manager*. Process-specific functionality and current status is provided by the *Process Manager*. *User Manager* provides particular access rights for any actions within the manager. For storage and retrieval of manager-specific data *Content Manager* is used for content repository functions.

Rule Manager

Short description	Mostly responsible for creation of fact templates, facts and rules from CCD, which are then part of simulation models.
Expected functionality	User interface for creation of simulation model parts with connection to CCD and particular data.
Input/output components	Simulation Manager, Content Manager, Version Manager, Concept Manager, Link Manager, User Manager, Process Manager, Search Manager.

Table 40 Rule Manager

This manager is responsible for one part of the process, where from CCD (parts of the) simulation models are extracted. User interface for evidence-based rules/agents creation and backward understanding of current modelling status are some of the main functions. *Process Manager* is used for process-specific operations and information. Search within the relevant resources (rules, agents, models, concepts in CCD, links, etc.) is supported by the *Search Manager*. Relevant objects are reused and managed also within *Concept Manager* and *Link Manager*. *User Manager* provides particular access rights for any actions within the manager. For storage and retrieval of manager-specific data

Content Manager is used for content repository functions, versioning support is provided by *Version Manager*.

Simulation Manager

Short description	Responsible for running simulation models and providing results of the simulations.
Expected functionality	Import/revision of simulation models, running simulations (in different modes), evidence-based inspection of models from rules to CCD and documents according to simulation results, creation of results (text-based, statistics, etc.).
Input/output components	Rule Manager, Concept Manager, Link Manager, Process Manager, User Manager, Content Manager, Version Manager, Document Manager.

Table 41 Simulation Manager

This manager is responsible for simulation part of the process, where simulation models are used in simulations and results of simulation are analysed. Simulation models can be imported and revised with the help of *Rule Manager* and evidence-based inspection using *Concept Manager* and *Link Manager*. *Process Manager* is used for process-specific operations and information. *User Manager* provides particular access rights for any actions within the manager. For storage and retrieval of manager-specific data *Content Manager* is used for content repository functions, versioning support for models is provided by *Version Manager*. *Document Manager* is used also for storage of documents created within simulations (results – text-based, statistics, etc.).

Search Manager

Short description	Component which provides (federated) search within the OCOPOMO system or partial searches in different resources (where needed).
Expected functionality	Provides search in particular resources (documents, forums, CCDs, etc.), individually or as a federated search, with or without tags, versions, process-specific information, etc.
Input/output components	Document Manager, Collaboration Space Manager, Discussion Forums Manager, Chat Manager, Calendar Manager, Annotation Manager, Concept Manager, Link Manager, User Manager, Process Manager, Content Manager, Rule Manager.

Table 42 Search Manager

This manager provides widely shared functionality within the OCOPOMO system – search for different type of data, metadata, content objects, etc. It is possible to combine more searches into one output or use partial searches within specific components (Document Manager, Collaboration Space Manager, Discussion Forums Manager, Chat Manager, Calendar Manager, Annotation Manager, Concept Manager, Link Manager, etc.). *User Manager* provides particular access rights for any actions within the manager. *Process Manager* is used for process-specific operations and information.

For storage and retrieval of manager-specific data *Content Manager* is used for content repository functions.

Collaboration Space Manager

Short description	Responsible for managing collaboration space where all communication aspects (sharing of documents, forums, chat, polls, etc.) are connected and shared together using one shared space.
Expected functionality	Creation of collaboration space (CS), management of members in CS, adding objects into space (by specific tools), opening objects within space, preference-based starting view of CS, and search in CS, notification features within space (e.g. RSS, hints, news, etc.).
Input/output components	Document Manager, Process Manager, Discussion Forums Manager, Chat Manager, Calendar Manager, Polling and Rating Manager, Notification Manager, Search Manager, Content Manager, User Manager.

Table 43 Collaboration Space Manager

This manager provides shared collaboration space within the OCOPOMO platform, especially communication utilities (*Discussion Forums Manager*, *Chat Manager*, *Calendar Manager*, *Polling and Rating Manager*), sharing documents (*Document Manager*) and management of members of shared space. *User Manager* provides and sets up particular access rights for any actions within the manager. *Process Manager* is used for process-specific operations and information. For storage and retrieval of manager-specific data *Content Manager* is used for content repository functions. Notification functionality, provided by *Notification Manager*, is used within collaboration space for publishing news, hints and providing RSS feed(s). Search within the collaboration space elements, relations and other relevant objects are supported by the *Search Manager*.

Notification Manager

Short description	Responsible for notification services within the OCOPOMO system, where any necessary information (by other components of collaboration space) could be provided using selected channel(s) like news, hints, newsletter, RSS feed inputs or email.
Expected functionality	Preparing notification message, selecting of channel(s) and execution of notification service.
Input/output components	Document Manager, Process Manager, Collaboration Space Manager, Discussion Forums Manager, Calendar Manager, Polling and Rating Manager, User Manager.

Table 44 Notification Manager

This part of the system is mainly responsible for providing its own functionality – preparing notification messages, selecting channel(s) and publishing (executing notification service) – to other components (e.g. Document Manager, Collaboration Space Manager, Discussion Forums Manager, etc.). *User Manager* provides and sets up particular access rights for any actions within the manager. *Process Manager* is used for process-specific operations and information.

Process Manager

Short description	Responsible for managing workflow of the whole process for policy modelling in OCOPOMO (from initiation through scenario building and analysis to modelling and simulation).
Expected functionality	Control of current process status within policy modelling process, changing steps within process, responsible for sharing process-specific data with other components for issues like process status, necessary requirements for finishing/changing current step, process-specific changes of access rights for users, notification of process-specific changes, process steps versioning.
Input/output components	Collaboration Space Manager, User Manager, Notification Manager, Content Manager, Version Manager, Annotation Manager, Rule Manager, Simulation Manager, Concept Manager, Link Manager, Search Manager.

Table 45 Process Manager

This manager is responsible for managing process steps for the OCOPOMO process of policy modelling. It is important for controlling process current status, changing steps, sharing data regarding process (*Collaboration Space Manager*), notification of changes (*Notification Manager*), reuse/sharing of process-specific data and information in all other components (*Annotation Manager*, *Rule Manager*, *Simulation Manager*, *Concept Manager*, *Link Manager*, *Search Manager*, etc.). *User Manager* provides and sets up particular access rights for any actions within the manager. For storage and retrieval of manager-specific data *Content Manager* is used for content repository functions. *Version Manager* is used for support of versioning functionality for process steps and resources.

Concept Manager

Short description	Responsible for managing structured conceptual descriptions of modelled problem (CCD).
Expected functionality	Creation of CCD elements and updating current structure, definition of metadata templates, storage, retrieving and visualisation of descriptions, export specific formats, versioning descriptions.
Input/output components	Content Manager, Version Manager, Process Manager, Link Manager, Annotation Manager, Rule Manager, User Manager, Simulation Manager, Search Manager.

Table 46 Concept Manager

This manager is responsible for managing structured information about the currently modelled problem known as CCD (conceptual descriptions which leads to simulation models). Basic functionality includes creation of CCD elements, updating structure, definition of structure of data, as well as storage, retrieval, visualisation and versioning them. *Annotation Manager* is using concept creation for explicit identification of structure in scenarios and data. *User Manager* provides and sets up particular access rights for any actions within the manager. *Process Manager* is used for process-specific operations and information. For storage and retrieval of manager-specific data *Content Manager* is used for content repository functions. *Version Manager* is used for support of versioning functionality for concepts in CCD. *Link Manager* and *Rule Manager* are directly connected to this structure information, therefore are also important and communicate with this manager. Search within the relevant resources (concepts in CCD) is supported by the *Search Manager*. Concepts are also available to *Simulation Manager* for evidence-based analysis of simulations.

Link Manager

Short description	Responsible for linking the evidence-based information within the OCOPOMO process.
Expected functionality	Creation of links (connection of information elements) between document parts and CCD elements, CCD elements and rules/simulation models, as well as extracted simulation results. Possibility to retrieve link connection structure between selected elements in order to achieve evidence-based explanations and better understanding of simulation results, scenario analysis and policy modelling issues.
Input/output components	Content Manager, Version Manager, Process Manager, Concept Manager, Annotation Manager, Rule Manager, User Manager, Simulation Manager, Search Manager.

Table 47 Link Manager

This manager is responsible for managing links between information about the currently modelled problem from evidence-based and CCD elements to rules, agents and simulation models (and their results). Basic functionality includes creation of links and retrieving of them for explanations in simulation analysis (starting from *Simulation Manager*) and policy modelling updates. *Annotation Manager* is needed for connection of concepts within structured data (as a part of the evidence-based linkage). *User Manager* provides and sets up particular access rights for any actions within the manager. *Process Manager* is used for process-specific operations and information. For storage and retrieval of manager-specific data *Content Manager* is used for content repository functions. *Version Manager* is used for support of versioning functionality for links. *Concept Manager* and *Rule Manager* are directly connected to this linking information, therefore are also important and communicate with this manager. Search within the relevant resources (links between concepts in CCD, data, agents, rules, etc.) is supported by the *Search Manager*.

User Manager

Short description	Responsible for managing users, roles, profiles and their access rights within the OCOPOMO platform.
Expected functionality	Creation of users, managing their roles and profiles, managing access rights (with direct connection to current process status), providing access rights to other components.
Input/output components	Process Manager, Annotation Manager, Document

	Manager, Discussion Forums Manager, Chat Manager, Calendar Manager, Polling and Rating Manager, Simulation Manager, Rule Manager, Collaboration Space Manager, Search Manager, Concept Manager, Link Manager, Notification Manager.

Table 48 User Manager

This part of the system has several main functions like creation of new users, managing their roles and profiles, and managing access rights with their provision to other components. All components from the first two levels are connected to this manager (see 'Input/output components' sections) and reuse user-specific information (access rights), *Process Manager* is most important for changing access rights during the process execution. Access rights are not needed on the bottom level (content repository and versioning), because these are not used directly by users (but components) and access rights have already been checked by caller functions of particular components.

Content Manager

Short description	Responsible for managing specific content (like conceptual descriptions, metadata for search, specific types of documents, links between conceptual elements, etc.) within the OCOPOMO platform.
Expected functionality	Directly managing necessary content storage and retrieval from the repository for different purposes (creation of conceptual descriptions, links, metadata, search in different content, managing of process-specific data) and types of resources, usage of versioning support (through Version Manager).
Input/output components	Document Manager, Version Manager, Collaboration Space Manager, Discussion Forums Manager, Calendar Manager, Polling and Rating Manager, Annotation Manager, Concept Manager, Link Manager, Rule Manager, Simulation Manager, Process Manager, Search Manager.

Table 49 Content Manager

This manager is responsible for managing content storage and retrieval from the content repository for different types of data, which are necessary within the OCOPOMO platform. Therefore many components are using content repository for storing and retrieving of such specific data like links, concepts, metadata, searches, process-specific data, etc. (see 'Input/output components' sections). *Version Manager* is used for the support of versioning functionality for content repository elements.

Version Manager

Short description	Responsible for versioning different data resources within the OCOPOMO platform.
Expected functionality	Versioning content repository objects of different types, updates of new versions, managing versions of content repository resources, use of versioning for other

	components.
Input/output components	Content Manager, Document Manager, Process Manager, Concept Manager, Link Manager, Rule Manager, Simulation Manager.

Table 50 Version Manager

This part of the system is responsible for versioning different data resources. The main responsibility is to support Content Manager and its content repository with versioning support. Moreover, several components are able to use versioning directly for their elements (if it is needed) like process-specific data (*Process Manager*), documents (*Document Manager*), concepts and links for structured CCD and semantic links between evidence-based data and policy modelling elements (*Concept Manager*, *Link Manager*), as well as versioning of simulation models and their parts (*Simulation Manager*, *Rule Manager*).

6.2. INFORMATION VIEW

The information view describes the means of storage, maintenance, and distribution of information through the system architecture [Rozanski and Woods, 2005]. It particularly includes the identification of expected information types, data sources and their mutual relationships within the system, analysis and specification of nature, content, structure, and ownership of the data produced or consumed within the system.

The design of information view is based on the analysis of user requirements, which indicates an initial distribution of information resources and data flows. Considerations of the selected technology platform, design approach, and expected information types scaffold the information structure from a global perspective. The data architecture, based on the analysis and assumptions, includes the specification of data objects and their relations, information flow of dynamic data transformations, as well as the scope of data access for particular actors during the system runtime.

User requirements were specified in [Bicking et al., 2010] mostly from usability and functionality perspectives. However, practically all the requirements imply a presence of some data structures and information resources, which should be properly represented in the architecture design. A detailed analysis of provided user requirements towards the specification of information resources and data objects is presented in Appendix D.

6.2.1. Design considerations

Web-based client-server application

The OCOPOMO system will be designed as a web-based application (see, for example, the requirement I-1 [Bicking et al., 2010]), which typically consists of three layers: portal-based user interface, business logic, and data layer. The design provided in the information view of architecture will be focused on the data layer structures and the respective objects (components, information resources) will be proposed for the business logic layer.

Object-oriented design

The design of the data architecture will follow the principles of the object-oriented design [Coad and Yourdon, 1991]. This advantageous approach is almost exclusively used for client-server applications, because it enables a logical separation of particular system components, their relations and interfaces. The basic design element is a data object, which is characterised by a set of properties (attributes). Data objects are related to each other by data transformation interfaces. The properties of objects can be inherited, so that the data objects can be organised in a generalisation hierarchy.

Structured vs. unstructured data

The distinction of the data structure affects the representation physical storage of data objects. Structured or semi-structured data can be represented and stored in (relational or object-oriented) databases, XML-like files or ontologies, and can be retrieved effectively by a standardised query language. Unstructured data such as documents and various multimedia files need to be stored and managed in special repositories. Their maintenance often includes text or metadata pre-processing, heuristic parsing, and indexing of the data content. The unstructured information is then available by querying the index by means of a specialised query language or interface.

Application-specific data vs. meta-data

According to the proposal provided in section 3.5.2, the OCOPOMO platform will be built upon the Alfresco / Alfresco Share framework and will reuse several third-party systems or components for partial functionality of policy modelling, scenario generation, and e-participation. The application-specific data are already defined in these components and the overall data architecture should reflect them. Contrary, the meta-data are under full control of the OCOPOMO system designers and can also be used to customise the third-party components by adding some information specifically required for the OCOPOMO platform.

6.2.2. Overall data architecture

The identification of information resources and respective data objects is mainly based on the requirements defined by OCOPOMO user partners in [Bicking et al., 2010]. According to this analysis, the following types of basic information resources have been identified:

- CSM-IR: Content and semantics management;
- ePart-IR: e-Participation objects and tools;
- NS-IR: Narrative scenarios and related CCD;
- SM-IR: Simulation models;
- UMS-IR: User management and security;
- DR-IR: Centralised data repository.

Each of these information resources can be further divided into a set of data objects - elementary building blocks of the data architecture. The design of data objects was initially accomplished during the analysis of user requirements analysis; then it was enhanced and updated according to the findings provided in system functionality descriptions and use cases. The resulting schema of information resources, data objects and their structural correlations is depicted in the following figure. The description of data objects for particular information resource types is presented in the following paragraphs of this section.

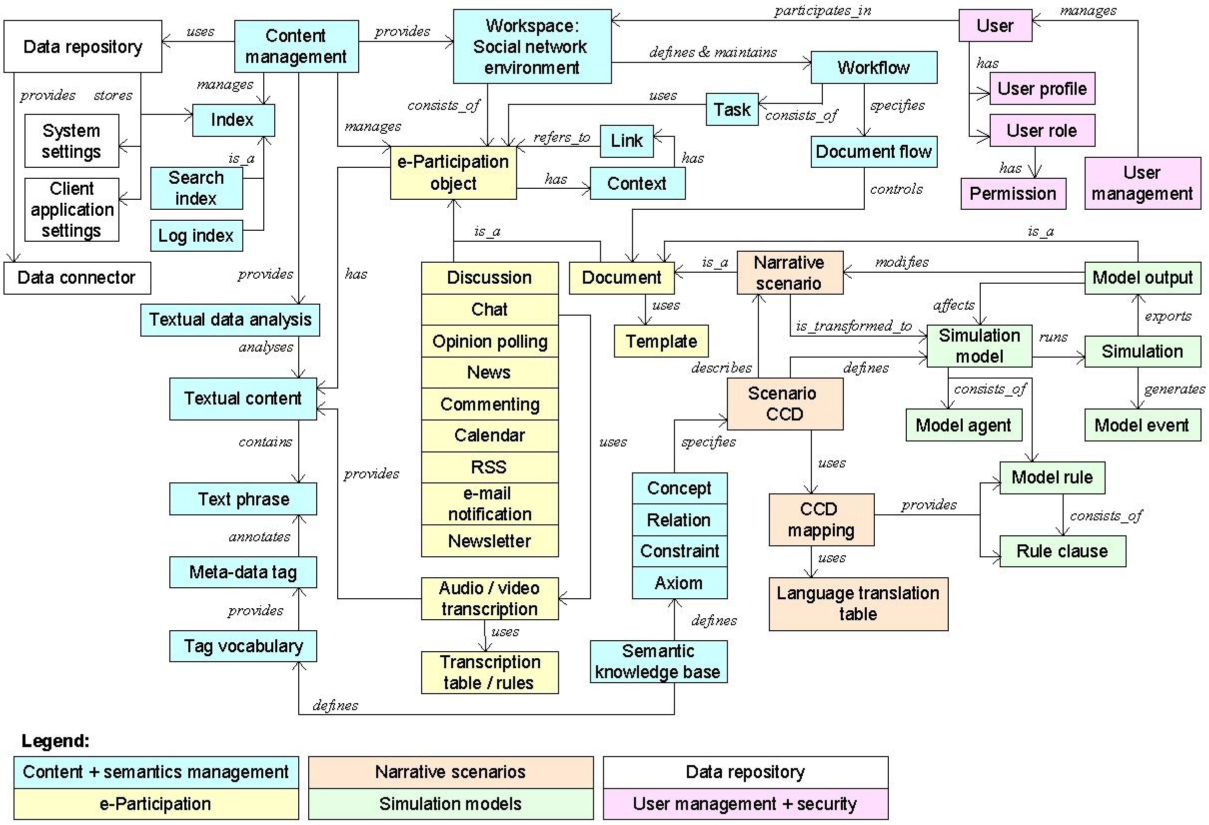


Figure 22 Architecture of information resources and data objects

CSM-IR: Content and semantics management

The CSM-IR represents data structures and resources that handle the collaboration process of policy modelling (i.e. the workspace - social network environment, workflow and document flow sequences), store, manipulate and semantically enhance the textual content of artefacts (i.e. index, textual data analysis, semantic knowledge model, context, etc.). The CSM-IR is composed of the following data objects:

- *Content management*: a general system for storage and maintenance of all artefacts required for the OCOPOMO functionality. Provides a technical environment for storage, version control, access, and publishing documents and other materials produced during the policy model development; provides and manages e-participation tools and related objects; provides the social network environment for collaborative work of involved system users.
- *Workspace - social network environment*: a collaborative space of e-participation objects and social network structures, which is shared by involved users. The workspace, which allows sharing ideas, materials, and information between users, forms a collaborative environment focused on the development of a suitable policy model (i.e. described in corresponding scenario and supported by respective simulations).
- *Workflow*: a representation of pre-defined workflow sequences (or a more complex structures) of tasks and activities performed by involved users on various e-participation objects. The goal of the process represented by the workflow is to organise steps and actions towards successful collaborative development of the policy model.

- *Task*: a data representation of a single action, which a user (process actor) performs on an e-participation object; may include various task properties such as required inputs or preconditions, produced outputs or effects.
- *Document flow*: a structure that describes the exchange of documents (supporting materials) between the actors involved in particular tasks of the collaborative process.
- *Context*: a representation of a structure of relations (association links) between two e-participation objects or their consequents - for example, between discussions and documents, narrative scenarios and documents, scenarios and models, etc.
- *Link*: a (one-way) relation between two e-participation objects or their consequents (i.e. data objects inherited from e -participation objects).
- *Textual data analysis*: data structures for text parsing, analysis, and annotation (tagging).
- *Textual content*: a text extracted from e-participation objects, represented in some of the supported formats (preferably HTML, optionally TXT, PDF, or DOC/DOCX).
- *Text phrase*: a portion (fragment) of text with a specified location and length. The phrase can be a paragraph, sentence, one word or sequence of more words. Text phrase may be annotated – semantically described by metadata tags, selected from a tag vocabulary.
- *Metadata tag*: a semantic element that describes the meaning of a text phrase.
- *Tag vocabulary*: a structure of semantic elements, specifically designed for an annotation of text portions extracted from e-participation objects.
- *Semantic knowledge base*: underlying structure of semantic objects (concepts, relations, etc.), which is capable to conceptually describe a domain of interest. According to the domain complexity and capabilities required for building CCDs and/or creating text annotations, the knowledge base can be represented as a simple hierarchy of concepts or may be organised into a more advanced structure of topic maps or ontologies.
- *Concept, Relation, Constraint, Axiom*: a set of semantic objects of the knowledge base.
- *Index*: a generic data entity that provides storage and access means for structured data.
- *Search index*: a representation of full-text index of the textual content and meta-data of stored e-participation objects. The search index provides a standard query language for quick and effective retrieval of stored documents.
- *Log index*: a representation of the repository of log records generated by the OCOPOMO system. The log index provides a query language for filtering and retrieving stored data according to the input criteria.

ePart-IR: e-Participation objects and tools

The ePart-IR represents the data, objects and resources, which facilitate sharing materials, active communication and information exchange between OCOPOMO users in a collaborative environment. The ePart-IR is composed of the following data objects:

- *e-Participation object*: a generic and abstract data entity that encapsulates basic properties for particular types of consequent data objects as document, discussion, etc. The properties include a presence of textual content that may be annotated by metatags, availability of contextual links, integration into workflow tasks and document flow sequences.
- *Document*: a single document persistently stored in the OCOPOMO system. The documents are considered to be multimedia files with the content of various formats of the MIME

standard[127]. Physical storage of documents is maintained by the content management: the content of documents is stored in a file structure, accompanied with properties, document versions, search indexes, context links and metadata annotations.

- *Template*: a representation of a pre-defined pattern, according to which a document is created.
- *Discussion*: a complex data entity that includes structures for discussion forums, topics, threads, and contributions.
- *Chat*: a representation of data structures for on-line communication.
- *Opinion polling*: a complex data structure for opinion polling, questions and available question types, answers and reports on polling results.
- *News*: a representation of data structures for news messages.
- *Commenting*: a data object representing comments on e-participation objects (i.e. documents, scenarios, etc.) or their text portions.
- *Calendar*: data structures for messages organised in date/time sequences (i.e. calendar events).
- *RSS*: a representation of data structures for RSS feed, i.e. for publishing frequently updated content in a standard XML-based format.
- *e-mail notification*: a complex data structure for composition and distribution of e-mail messages to the dedicated users, which is usually invoked by an event detected in the system (for example, a new task was started in the workflow, a new version of scenario or simulation model was generated, etc.)
- *Newsletter*: a data structure for composition and distribution of periodic messages to the users, according to their preferences (i.e. daily / weekly / monthly, etc.). The newsletter is usually published on the portal and, in parallel, distributed via e-mail.
- *Audio / video transcription*: a representation of means for transcribing the audio / video sequences, produced by chat or similar multimedia communication channel, into written texts.
- *Transcription table / rules*: a schema or data structure for transcribing the audio / video sequences into the format of written text.

NS-IR: Narrative scenarios and related CCD

The NS-IR covers the data and resources for manipulating with narrative scenarios and related CCD structures, which are then used for transforming the scenarios to simulation models. The NS-IR is composed of the following data objects:

- *Narrative scenario*: a generic data object that represents narrative scenarios. It is designed as a sub-type of the Document object, which implies that the scenario has its textual content, may have a context defined, can be included into a workflow, etc. Moreover, the scenario can be transformed to a simulation model and can be modified by a simulation output, which is represented by the Model output object.
- *Scenario CCD*: a structure of semantic objects (concepts, relations, etc.) that formally represents the textual content of a narrative scenario. After the translation enabled by the CCD mapping object, the CCD can be transformed to a respective simulation model.
- *CCD mapping*: a translation schema that allows converting a CCD structure to rules and clauses of a corresponding simulation model. Information on the text phrases that were transformed to the respective model parts is preserved during the transformation.

[127] MIME Media Types: http://www.iana.org/assignments/media-types/

SM-IR: Simulation models

The SM-IR represents the data and resources related to the creation and modification of simulation models, including simulation games and outputs generated by a model. The SM-IR consists of the following data objects:

- *Simulation model*: a data structure of a simulation model, which is a simplified abstract view of the complex reality, logical representation of objects, phenomena, and processes. The data structure of the simulation model includes properties such as content, expressed as a set of rules (i.e. rule-dependency graph), state, versions (i.e. environmental aspects of the descriptive scenario), assumptions (minimal set of assumptions the model should carry), etc.
- *Model agent*: a representation of an active entity that participates in a simulation model. The behaviour and constraints of agents are described in the model by a structure of rules. System users, according to their roles and permissions, may play a role of one or more agents in running simulations.
- *Model rule*: a logical expression, usually in the IF-THEN-ELSE form, included in the content of a simulation model. The rule consists of an ordered set of clauses.
- *Rule clause*: a logical object, part of the rule of simulation model. The clause may be derived from the scenario CCD and thus it may be related to particular text fragment in the underlying narrative scenario.
- *Simulation*: a representation of a "model in action", i.e. a simulation model running with given agents, upon specified inputs, parameters and conditions. The simulation parameters may include, for example, a desired level of details, time scale, number of cycles, etc.
- *Model event*: a representation of events generated by the model during a simulation.
- *Model output*: a document, which is generated by the running simulation as output. The generated document can be used as a specific model-based scenario, which then may affect changes in the original narrative scenario and/or in the related CCD.

UMS-IR: User management and security

The UMS-IR provides structures for user management, authentication and authorisation data. It includes user roles, profiles and preferences, access rights, credentials, etc. The UMS-IR is composed of the following data objects:

- *User management*: a repository for all the data necessary for maintenance of system users and their accounts. It, for example, includes a structure of users (given by properties, roles, or user types), data for identity management, credentials and access rights, etc.
- *User*: a generic data object that represents particular actors interacting with the OCOPOMO system, including the settings (properties) for user profiles, roles, and permissions.
- *User profile*: a storage place that contains user preferences (i.e. a customisation of the *Workspace*, frequency of receiving newsletters or notifications, etc.), identification and personal data, as well as an individual work of the user (i.e. private documents, temporary or experimental models, alternative scenarios, etc.).
- *User role*: specifies rights, permissions, and competencies that the user may possess during his/her interaction with the system (in the scope of a single *Workspace*). Available user roles correspond to the actors identified previously (i.e. Facilitator, Modeller, Analyst, Administrator, etc.). A list of corresponding permissions is provided as property of the user role.

- *Permission*: an allowance to perform particular action within the system, including proper authorisation settings and credentials. Permissions are instantiated to the action types provided by other OCOPOMO modules – for example, to moderate a discussion, to receive e-mail notifications, to comment a scenario, to modify a simulation model, etc.

DR-IR: Centralised data repository
The DR-IR represents an integrated data storage place for the whole web-based system. It consists of the following data objects:

- *Data repository*: a system of databases and file-based repositories, which stores the persistent system data and provides an effective data access.
- *Data connector*: an object that provides for inner system components the means for addressing, accessing, and retrieving proper data from the repository. Typically, it is the connection string for relational databases or the root path for file-based repositories.
- *System settings*: global settings and environment properties such as e.g. supported language versions, paths and URLs, location of client applications, etc. for a single installation of the whole OCOPOMO system.
- *Client application settings*: the configuration data for web-based client-side tools and applications of the OCOPOMO system. At the current state of the project, we can identify the OCOPOMO administration tool for the overall system maintenance and the client-side interface providing all the policy modelling functionality for dedicated users (see Figure 23).
- The configuration data may then include such information as, for example, location of data resources, mode of operation, etc.

6.2.2.1. Information flow

The information flow, as a part of the architecture design, represents an information and data exchange between main system components. The design of the OCOPOMO information flow is based on the assumptions of a client-server solution built on the Alfresco framework, as well as on the above-presented functional architecture (cf. section 6.1.2 and Figure 21) together with the distribution of information resources and data objects within the system. Proposed information flow is schematically depicted in the following figure.

The schema of information flow was constructed as a mapping of designed information resources into the proposed structure of functional modules, with respect to the architecture of Alfresco framework. The design of information flow follows the concept of three layers, distinguishing the data layer, inner business logic, and user interface. The DR-IR data objects are managed on the data layer, while the rest of information resources belong to the layer of business logic.

The information flow starts at the data layer, where the DR-IR data are stored in several repository types. Alfresco already provides the MySQL database for structured relational data, the file system storage for files, and the Lucene search index (http://lucene.apache.org). In addition, Alfresco contains the Hibernate framework for accessing the data on an object level. These technologies will be customised and implemented in the Version and Content managers, which then, via the data repository API, provide the means for full data access and manipulation for other components on the business logic layer.

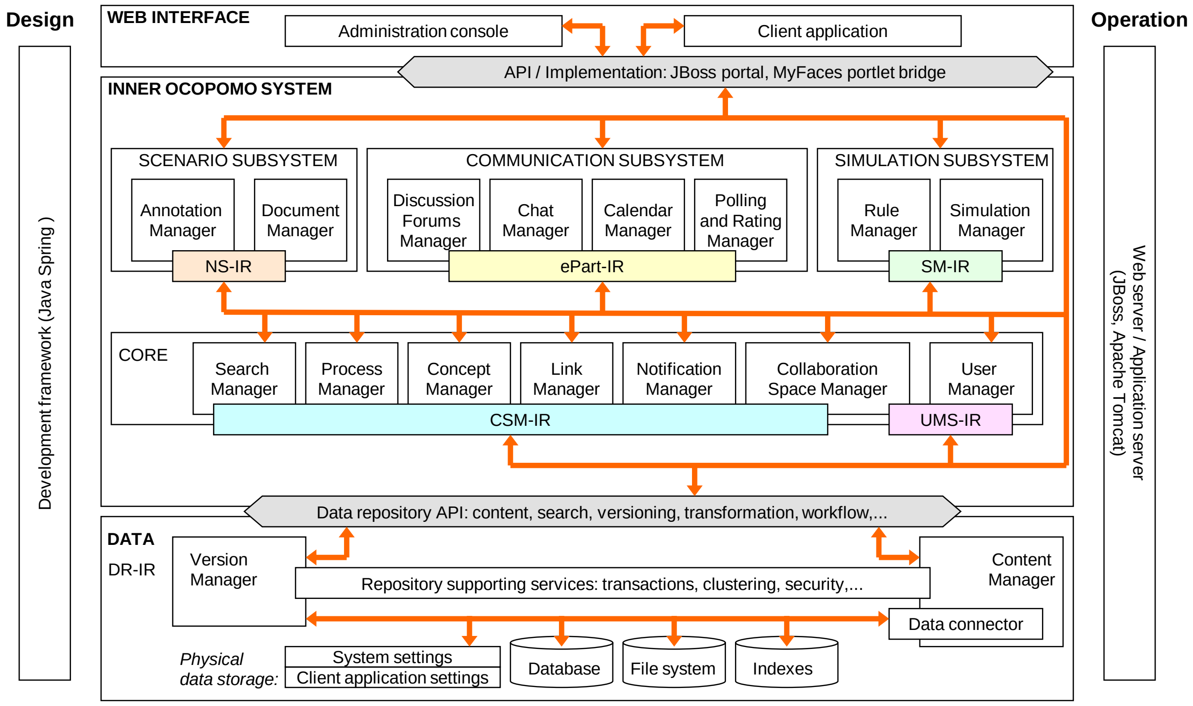


Figure 23 Information flow in the OCOPOMO system

The distribution of particular information resource types to the respective functional components, as depicted in the information flow schema, is rather rough. The reason is that the data elements are tightly correlated across information resource types, as it is presented Figure 22. Moreover, the components usually need to manipulate with the data objects of more than a single resource type. So the schema Figure 23 depicts typical relationships between components and information resources only. The components in the Core subsystem work mostly with the CSM-IR and UMS-IR resources. The User manager maintains the UMS-IR data, transforms and provides it to other components in a suitable object representation. The Collaboration space manager typically uses both of these resources and connects the workspace - social network environment data to the dedicated users. The data objects of NS-IR, ePart-IR, and SM-IR are consumed, handled, transformed and provided by the components of Scenario, Communication, and Simulation subsystems, respectively. The processed data are then forwarded to the system API, which is implemented in Alfresco by the JBoss portal technology. The resulting web content is presented to users in a web-based interface, which can be the system administration console or a specific OCOPOMO client application. The information flow is, however, bi-directional – users may do some actions and/or enter the data into the web-based interface; the data are then propagated to the components of inner subsystems and consequently, after proper transformations, are stored on the data layer.

6.2.2.2. Data ownership

The data ownership is a distribution of responsibilities and permissions between various types of external users during their interactions with the OCOPOMO system. The main information resources and data objects affected by particular types of actors / user roles are summarised in Table 51. The mode of data transfer (i.e. C/U/D/R - Create / Update / Delete / Read) is indicated in the last column.

Actor	Affected information resources	Mode
Politician	NS-IR: Narrative scenario, Scenario CCD	U/R
	CSM-IR: Workspace – social network environment	U/R
	ePart-IR: Document, Discussion, Chat,...	C/U/D/R
	SM-IR: Simulation model	U/R
Civil servant	NS-IR: Narrative scenario, Scenario CCD	U/R
	CSM-IR: Workspace – social network environment	U/R
	ePart-IR: Document, Discussion, Chat,...	C/U/D/R
	SM-IR: Simulation model	U/R
Stakeholder	NS-IR: Narrative scenario, Scenario CCD	C/U/D/R
	CSM-IR: Workspace – social network environment	U/R
	ePart-IR: Document, Discussion, Chat,...	C/U/D/R
	SM-IR: Simulation model	U/R
Facilitator	NS-IR: Narrative scenario, Scenario CCD	U/R
	CSM-IR: Workspace – social network environment	C/U/D/R
	ePart-IR: Document, Discussion, Chat,...	C/U/D/R
	SM-IR: Simulation model	U/R
Analyst	NS-IR: Narrative scenario, Scenario CCD	C/U/D/R
	CSM-IR: Workspace – social network environment	U/R
	ePart-IR: Document, Discussion, Chat,...	C/U/D/R
	SM-IR: Simulation model	U/R
Modeller	NS-IR: Narrative scenario, Scenario CCD	R
	CSM-IR: Workspace – social network environment	C/U/D/R
	ePart-IR: Document, Discussion, Chat,...	U/D/R
	SM-IR: Simulation model	C/U/D/R
Administrator	DR-IR: Data repository, System settings	C/U/D/R
	NS-IR: Narrative scenario, Scenario CCD	U/D/R
	CSM-IR: Workspace – social network environment	C/U/D/R
	ePart-IR: Document, Discussion, Chat,...	C/U/D/R
	SM-IR: Simulation model	U/D/R

Table 51 Data ownership for external actors interacting with the system

6.3. INTERNATIONALISATION PERSPECTIVE

Internationalisation is the process of designing software in a way that can be adapted to different languages and regions easily. Sometimes the term internationalisation is abbreviated as i18n[128].

An internationalised program has the following characteristics:

- With the addition of localized texts, the same executable can run worldwide.
- Textual elements such as status messages and the GUI component labels are not hardcoded in the program. Instead they are stored outside the source code and retrieved dynamically.
- Support for new languages does not require recompilation.
- Culturally-dependent data, such as dates and currencies, appear in formats that conform to the end user's region and language.
- It can be localized quickly.

Localization is the process of adapting software for a specific region or language by adding locale-specific components and translating text. Localization involves not only changing the language interaction, but also other relevant changes such as display of numbers, dates, currency, and so on.

The aim of the OCOPOMO platform is to be used by different stakeholders in different countries (with their own languages and country-specific settings). Therefore, the system should be independent from the language and specific settings partners might use (system should be able to support them automatically according to their basic internationalisation settings).

The main aspects that might influence the use of the OCOPOMO system by different partners are for instance languages spoken, different measurement units such as length, capacity, money or time slots. Internationalisation is not OCOPOMO's objective but we are aware that some issues might arise concerning this perspective.

Of course, from all internationalisation issues, most important one is multilingual interface (localization of interfaces). At least three languages in OCOPOMO have to be used for localization of platform. Two of them are pilot specific, Italian and Slovak, and one is a generally used language - English (in order to have at least one suitable interface for all other partners in the project, e.g. modellers, scenario analysts, developers, dissemination partners, etc.). According to these needs the OCOPOMO system will provide some mechanism for preparing all necessary resources. The main benefit should be a simple localization also for other languages (e.g. other languages of OCOPOMO partners like German and Polish). All localization stuff should be available using resource files, which are outside the code.

After the implementation deployment of certain components might need to include some code in order to deal with internationalisation. For instance, if Java applications are used for OCOPOMO, internationalisation can be simply achieved by adapting them as suggested in I18N Java tutorial[129].

6.3.1. Relevant user requirements

I-35	Multilingual interface

[128] There are 18 letters between the first "i" and the last "n".

[129] Trail: Internationalization (The Java Tutorial) - http://java.sun.com/docs/books/tutorial/i18n/index.html

I-NFT-8 Look and feel

6.3.2. Design considerations

Global vs. local perspective

Due to fact that different partners involved in the policy modelling process (at least in OCOPOMO pilot cases, where we have modellers and scenario analysts) could be from different countries, one of the issues that arise is the language problem. Even though the OCOPOMO system languages will be English, Italian and Slovak we should consider in the architecture whether different partners speak different languages. The OCOPOMO platform should consider a mechanism in order to make all the important system messages and notifications understandable and error prone for all the partners. This mechanism could allow partners to select a language which they would like to communicate with the rest of partners. Other issues that could arise are different time slots of partners involved in policy modelling process. The platform should perform automatic conversion of day-time and help users know working hours of corresponding partners. Also, the OCOPOMO platform should be able to automatically convert between local units of measurement (if users are interested), e.g. currency, length, capacity, etc.

6.3.3. Applications to relevant views

Functional view

Internationalisation could be considered when functionality features of OCOPOMO are dealt with.

Information View

This part of the system should be in charge of the date and time conversion and perhaps the language problem. In second case main objective is to prepare and store resource files for every created user interface according to i18n standard (using simple mechanism like in localization resources available in Java platform technologies).

6.4. INTERACTION PERSPECTIVE

The interaction perspective describes user interface views and users’ interactions possible with the OCOPOMO platform. Therefore mock-ups have been created (see Appendix C). The design of the mock-ups is based on the analysis of user requirements and use case diagrams and descriptions presented in section 5.

Design considerations, which have been taken into account when creating the mock-ups, are presented in the next subsection. Afterwards, the collaboration space of the OCOPOMO platform is introduced in section 6.4.2. Section 6.4.3 describes users’ interaction options to generate scenarios in a collaborative way. Mock-ups demonstrating policy modelling features are not part of this deliverable, as they are expected to be detailed in upcoming project deliverable D5.1 [Moss et al., 2010].

6.4.1. Design considerations

Web-based application

The OCOPOMO system should be designed as a web-based application (see, for example, the requirement I-1 in [Bicking et al., 2010]). As described in section 3.5.2, Alfresco is used for providing a web-based platform with content management facilities and collaborative means (e.g. discussion forums). The mock-ups are created with the help of an Alfresco test system. This helps to create mock-ups which show realistic views of the final platform. Therefore screenshots are taken from the test-system and additional features and functionalities are added into the screenshots.

Usability

Usability is an important factor for software to be accepted by provisional users. Regarding the interaction perspective, a smooth and intuitive transfer from one screen into the next is necessary – it is essential to simulate the final system and in this way mediate look and feel of the prospective system to users at this early stage of development. In order to ensure this, the mock-ups have been exported in the format of HTML files. Pushing the buttons guides users to the subsequent mock-up screen.

6.4.2. Collaborative space

The collaborative workspace is a virtual space where users can come together and contribute to the overall policy modelling process. The user starts at the main page of the project website (see Figure 63). The home page gives introductory information about the objectives of the website (on the home page), the OCOPOMO project (click the link OCOPOMO) and policy modelling and simulation (on “Model your future”) and provides links to its other projects, if there are public projects available. The home page is the first page seen by every visitor. A search functionality is available, too.

A log-in screen is available in the middle of the start page and on the right top corner of every page. The registered user can log-in with user name or e-mail. He can ask for his password by clicking on the “Forgot password” (see Figure 65) link or register by clicking on “Register” (see Figure 64).

6.4.2.1. Registration

If a user is invited into the collaboration space of a project, the user must first register. Notifying a new user of the collaboration space is referred to as registration. The user has to type in some basic information for registration. The registration is completed by pressing the “Register” button. The user is directed to the inner main webpage. From this inner main page, the user can follow all the links to other webpages under the same domain. Once the user has been registered, the registered user can access the content of the collaboration space according to the rights assigned to him/her.

It might happen that the user forgets his/her password. To obtain a new password, the user can click on the link “Forgot password” (see Figure 63) thereby the user is being forwarded to the password prompt webpage (see Figure 65). At this page he/she is requested to enter his/her e-mail address. Then either the new password is sent to the user or a link will be emailed to the user. After receiving the new password, the user can type in his/her access data at the home page (see top right corner of Figure 63) and access the collaborative space.

6.4.2.2. Dashboard

The Dashboard (see Figure 68) is the first screen a users sees regularly after log-in. It is a webpage which is custom-tailored to the individual user's preferences and shows the user the recent activities after the last log-in. This is important for a facilitator in particular. He can see the newest changes made by the other users.

The view is related to requirement I-F-16 which says that the system enables every registered user to create a personalized homepage, which will not substitute the existing portal homepage. Rather it serves as an aggregator of information, which is of the user’s interest.

More specifically, each user can select one or more sections from the portal’s available sections (i.e. forums, news, processes etc), in order to have an overview of the latest developments in these areas available in their own, personal homepage (The dashboard can be customized by clicking “Customise Dashboard”). The order of appearance of these sections can also be customized. Furthermore, they can select one or more of the available areas of debate and have content in their homepage that is related to these areas, highlighted in the colours of their choice. But the dashboard does not need to be customized as it has a usable structure from the beginning (given by its defaults).

The dashboard shows:

- “My Calendar” with the recent events, published in the platform and visible for the user.
- “Getting started” with a number of important information about the platform and its features. Only functionalities, which are helpful for the user, are visible. For example, if the user does not have the rights to add a project, the “Create a Project” section is not displayed. Otherwise it is available.
- “My Profile” shows the information, which is available about the user. By clicking on “View full profile” users can see, edit and delete their profiles.
- “My projects” shows all projects, which are visible to the user. This means that the box shows the projects, which the user has created or has been invited to.
- OCOPOMO Feed represents the latest news about OCOPOMO or the projects. Can be published by facilitators.

By clicking on the menu item “Projects” the user comes to a website, which shows the projects, which the user can access. The menu item “People” shows a list with the user names of all registered users (within the project) when clicked. The “Repository” shows all project related documents, which the user can access in the platform. Other functionalities are available under “More...”. A search functionality is available, too. The latest four functionalities are “standard” functionalities of Alfresco. Therefore they are not further explained.

6.4.2.3. News entry

Publishing recent news about e.g. the policy area, decision making process should be an ongoing process during the whole initiative. News in the portal should be rather short and up-to-date, sometimes linked with the other contents in the portal and provide important background information about the topics. Discussions can be started together with news published. The functionality to comment news (similar to blog functionality) supports the interactivity of the news section. No extra blog functionality is needed because external as well as project internal news can be presented with this kind of news functionality.

The requirements T-24, T-25 T-C2 [Bicking et al, 2010] are related to the news functionality. The requirement T-24 is changed in the way that news can also be restricted to a certain project/user group.

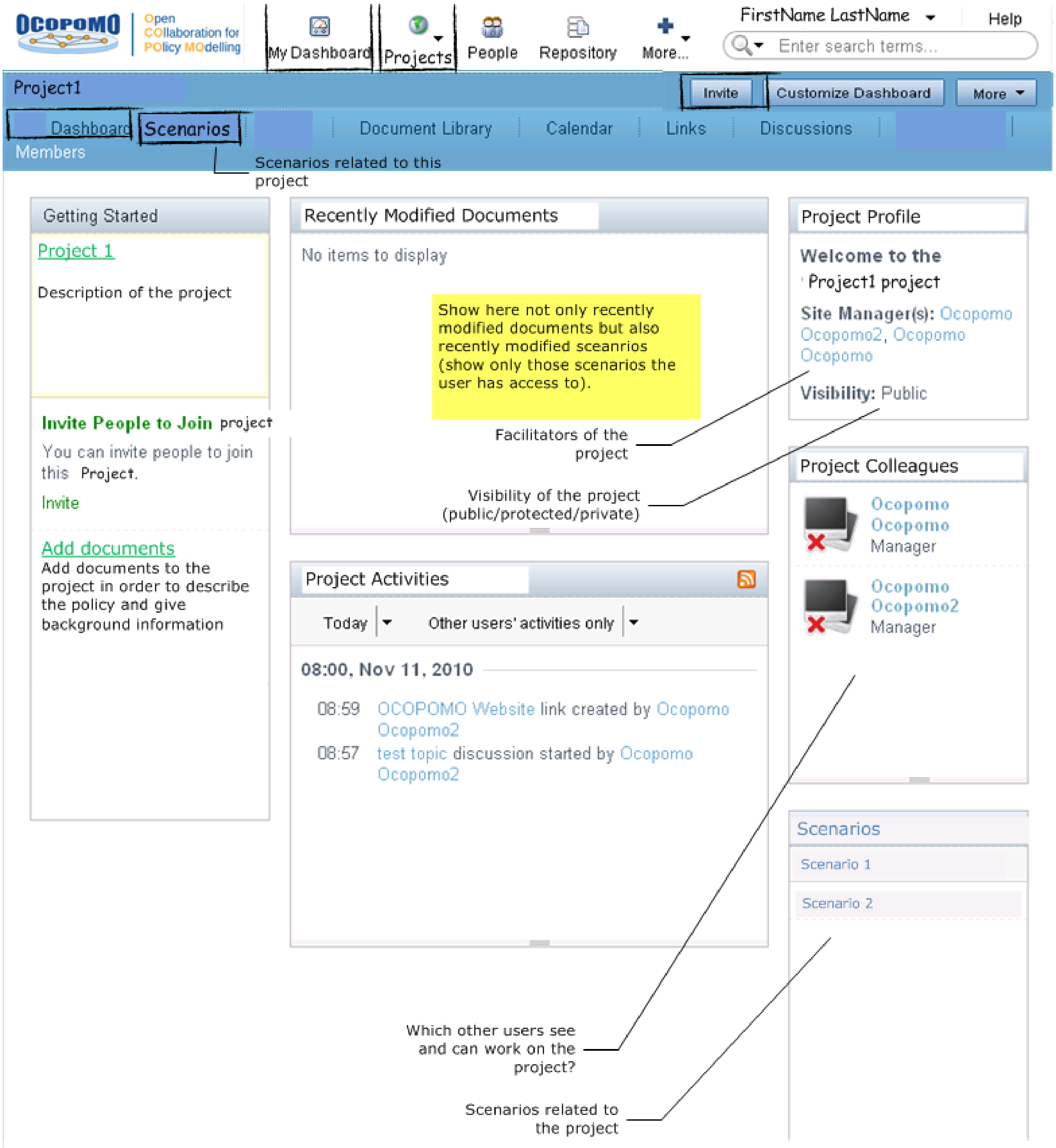
Figure 81 shows the mock-up for a news entry with commenting and rating functionality.

6.4.2.4. Project

A project is a particular area of the collaboration space, which is used in order to collaborate with other users to bring forward a policy issue. A project is created by a facilitator by clicking on the "Create Project" link (see Figure 68). The policy description of a project includes text descriptions and official background documents. The mock-up for creating a project is visualised in Figure 82. The user can type in a name and a description. After creating a project, a project dashboard is displayed as visualised in Figure 24. The "Getting Started" box shows activities, which are possible for the user. The authorized user can add documents and invite other people to join the project. The boxes "Recently Modified Documents" and "Project Activities" show the latest changes in the project. The menu item "Document Library" shows a list with all documents, which are linked with the project. A "Calendar" gives the facilitator the possibility to add events interesting for the users who are collaborating in the project. The granted users are able to add references to other interesting web sites when clicking the menu item "Links". A discussion forum is available when clicking the menu item "Discussions". The menu item "Members" shows the "project colleagues", i.e. who can read and/or edit the project. The project dashboard also shows the scenarios, which are related with the project.

After a project has been created, the initiator and the facilitators can invite users (including the invitation of registered and unregistered users, i.e. users who are not yet involved but who were identified as valuable contributors) to update the descriptions and to add new ones. Figure 83 shows a mock-up for the invitation screen. The facilitator can search for people (i.e. registered users), add external users (with first name, last name and e-mail) and assign them roles ("Set All Roles to").

The user can see an overview of all projects (cf. Figure 84) by clicking on the menu item "Projects". Facilitators have the possibility to delete projects. In addition, users have the possibility to leave projects or to join projects. Joining a project is possible in the case the project is public or the user has been invited to the project.



Scenarios related to this project

Project 1

Description of the project

Invite People to Join project

You can invite people to join this Project.

Invite

Add documents

Add documents to the project in order to describe the policy and give background information

No items to display

Show here not only recently modified documents but also recently modified sceanrios (show only those scenarios the user has access to).

Facilitators of the project

Visibility of the project (public/protected/private)

Welcome to the Project1 project

Site Manager(s): Ocopomo Ocopomo2, Ocopomo Ocopomo

Visibility: Public

Ocopomo Ocopomo Manager

Ocopomo Ocopomo2 Manager

08:00, Nov 11, 2010

08:59 OCOPOMO Website link created by Ocopomo Ocopomo2

08:57 test topic discussion started by Ocopomo Ocopomo2

Which other users see and can work on the project?

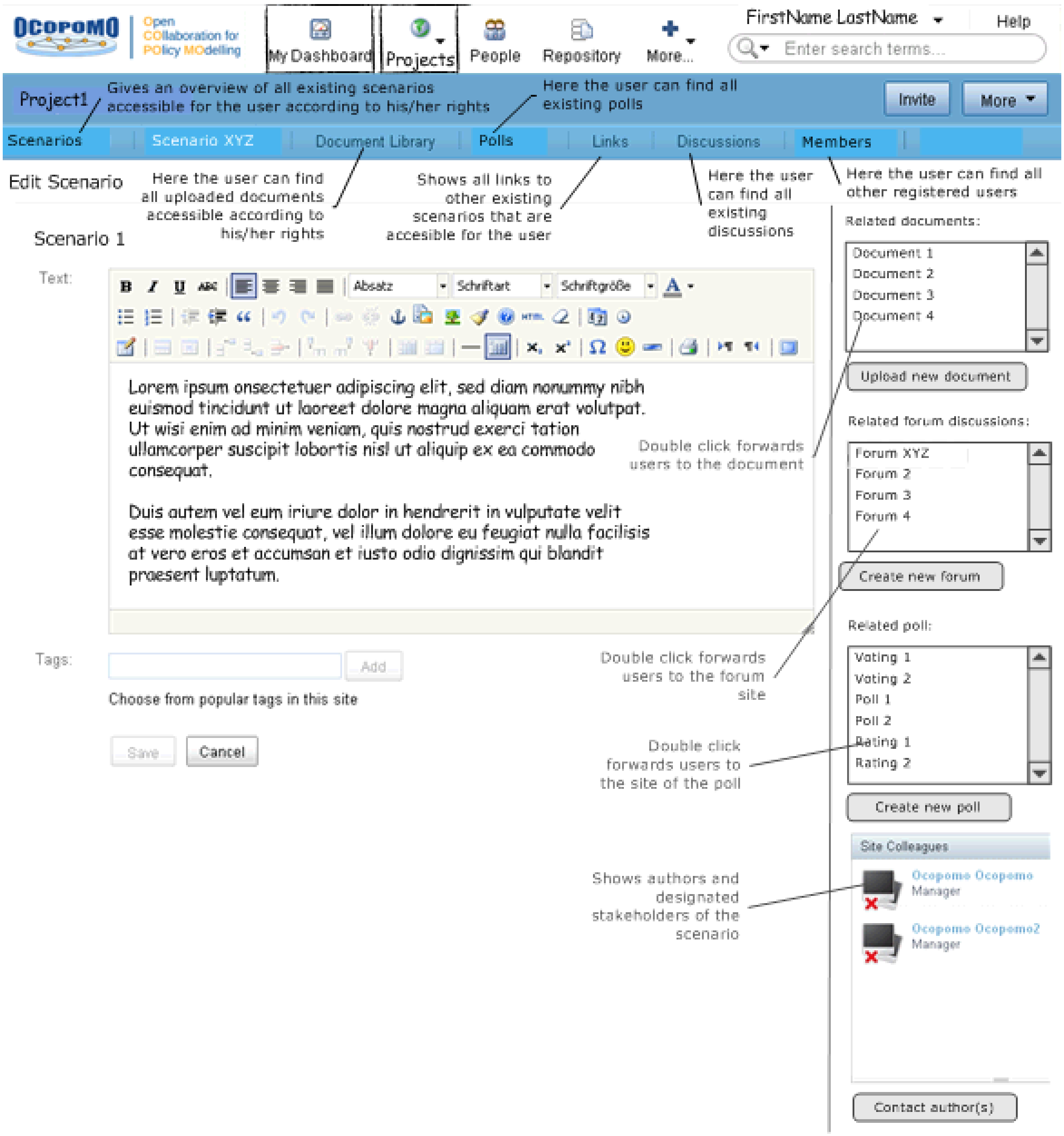
Scenarios related to the project

Figure 24 Mock-up of the project dashboard

6.4.3. Collaborative scenario generation

The collaborative scenario generation starts with viewing the list of existing scenarios (see Figure 72). At the beginning of the project only the initial scenario generated by the facilitators in conjunction with the initiators of the project will be available in the list. The more active the stakeholders are in generating scenarios, the more scenarios will be available in the list. Users of the system can view existing scenarios either by selecting a scenario from the list with all scenarios included or by searching for scenarios on the basis of tags (cf. Figure 72). The more tags are assigned to one scenario by the author(s) (see Figure 73 and Figure 25), the better the quality of the search is.

However, if a specific scenario is selected, the user receives the respective view on the scenario (see Figure 74). Then the user can decide if he/she wants to contribute to the scenario and if yes, in which of the variety of ways (see Figure 25).



Gives an overview of all existing scenarios accessible for the user according to his/her rights

Here the user can find all existing polls

Here the user can find all uploaded documents accessible according to his/her rights

Shows all links to other existing scenarios that are accesible for the user

Here the user can find all existing discussions

Here the user can find all other registered users

Lorem ipsum onsectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum.

Double click forwards users to the document

Double click forwards users to the forum site

Double click forwards users to the site of the poll

Shows authors and designated stakeholders of the scenario

Figure 25 Mock-up for editing scenarios

The user is able to edit the scenario by changing the description (i.e. adding new information, changing and manipulating existing parts or deleting information).

6.4.3.1. Document sharing

Several features are provided to facilitate the communication and information exchange with the intention of mutually learning and understanding. The system allows, therefore, for document management. The user can view scenario-related documents by a double click on the respective document (see right side menu in Figure 25, Figure 79). With it, the user will be directly forwarded to the document. Furthermore, the user is also able to add new documents to the scenario by clicking on the button “Upload new document” (see right side menu in Figure 25). Then the user will be directly forwarded to the site where he/she can upload and describe the document (see Figure 79). The user can also view all documents by clicking on the menu item “Document Library” (see main menu at the top of the site). Document related features shall not only inform the user but also allow for collaborative data gathering and information/knowledge sharing.

6.4.3.2. Discussion forum(s)

Communication features are provided to enable users of the project website to interact with each other by exchanging tips and discussing hot topics related to a scenario theme. Therefore, users can join an existing discussion forum or create a new forum (cf. Figure 77 and Figure 78). Forums save information posted on a particular topic for other users to see it at any time. This creates a discussion environment and allows the facilitators to gather further relevant information for policy modelling. Everything that gets posted gets read again and again. The fact that the discussion is not real time means that it rarely turns into heated arguments as users are given time to research and consider their comments before replying. This makes for mostly high-quality discussion. Each forum has a hierarchical tree-like structure, i.e. a forum consists of subforums, which again consist of topics (see Figure 77). A single conversation on a topic is called thread where users are able to reply (see Figure 78). The user can view scenario-related discussions by double click on the respective discussion (see right side menu in Figure 25). With it, the user will be directly forwarded to the discussion. Furthermore, the user is also able to create a new discussion based on a theme from the scenario by clicking on the button “Create new forum” (see right side menu in Figure 25). Then the user will be directly forwarded to the site where he/she can set up a new forum (see Figure 77). The user can also view all discussion forums by clicking on the menu item “Discussions” (see main menu at the top of the site).

Experiences from e-participation projects monitoring and evaluation showed that in most forums people who want to post have to register by giving their email address and names. This allows the moderator to follow up and email them in the future with information about the project (such as new services). These people seldom consider this contact as spam because they know the background.

6.4.3.3. Public opinion polling

The system offers public opinion polling as one way to obtain an unbiased view of the public opinion on a range of issues related to the policy theme. Polls can show trends in the concerns, fears and hopes of stakeholder groups, which are important for policy modelling exercise. The user can view all polls related to one scenario by double click on the respective discussion (see right side menu in Figure 25). With it, the user will be directly forwarded to the poll (whereby polls can be questionnaires, ratings, and voting depending on the purpose of the poll). Furthermore, the user is also able to create a new poll based on a theme from the scenario by clicking on the button “Create new poll” (see right side menu in Figure 25). Then the user will be directly forwarded to the site where he/she can set up a new poll. The user can also view all existing polls by clicking on the menu item “Polls” (see main menu at the top of the site).

6.4.3.4. Showing site colleagues and contacting author(s)

For collaborative scenario generation it might be helpful to build relationships with users and help them to build relationships with each other. Regular posts and active discussions among forum members can support this as they get to know each other. So, it is a nice feature of the system to show its users who are their site colleagues. Besides, some users may want to more personal exchange information on specific issues. Therefore, the system offers users the opportunity to contact the author(s) of the scenario (see Figure 75 and Figure 25).

6.4.4. Application to relevant views

Functional view

All aspects within the interaction view are already incorporated in current functional view (through use cases and user requirements) or they will be used (mock-ups) during the implementation of particular components and their user interfaces. Therefore we do not need to update functional view, if all recommendations from interaction perspective will be respected in the subsequent implementation phase.

Information view

No update is needed for the information model. All data-related aspects (relevant to interaction view) were added into the model during user requirements identification phase and use case modelling. Mock-ups are only implementation-related and are connected to use cases.

6.5. USABILITY PERSPECTIVE

Usability has multiple components, it is not a one-dimensional property of a user interface. Traditionally, it is associated with the following five usability attributes of the system: easy to learn, efficient to use, easy to remember, low error rate, and pleasant to use [Nielsen, 1993].

E-participation and collaboration services via electronic channels, as provided in OCOPOMO, need to be simple, effective, easy-to-use and functional. Besides this, the look and feel as well as the fun-factor should not be underestimated [Scherer et al., 2009]. Especially in contexts, where heterogeneous user groups should actively participate in policy discussions and participatory decision-making by electronic means, usability is crucial.

6.5.1. Usability engineering

To fulfil usability requirements, the design and implementation of the OCOPOMO platform should follow well designed processes. Systematic usability engineering is necessary at least to ferret out minor design details that influence usability [Nielsen, 1993]. Bad usability on local government web sites may destroy the strategy of the whole website [Esteves, 2007]. Therefore all decisions about the OCOPOMO user interface need to be the result of a systematic process and should be documented.

Usability engineering is not one single step in the product development cycle. It is a set of activities that should take place throughout the lifecycle of the product. Nielsen proposes a number of steps for the user engineering lifecycle [Nielsen, 1993, p. 72f]: (1) Know the user; (2) Competitive analysis; (3) Goal setting; (4) Parallel design; (5) Participatory design; (6) Coordinated design of the total interface; (7) Guidelines and heuristic evaluation; (8) Prototyping; (9) Empirical testing; (10) Iterative design; (11) Feedback from field use.

It is not always possible to perform all these steps in one product lifecycle [Nielsen, 1993]. There are a number of other lifecycles specialised and adapted for different project types [Mayhew, 1999].

In order to ensure the usability of the OCOPMO platform, it is recommended to have an iterative design and development process and a pilot evaluation phase performed before the official launch of the platform. The iterative design process means that the proposed solution will be tested at several levels against the requirements and usability goals considered in the requirements analysis phase using methods as heuristic evaluation and empirical testing. If the proposed solution does not meet the usability goals, the design will be improved. The iterative design and development process starts with the design of the architectural views, then goes beyond the pilot implementation, and ends with the launch of the platform.

One problem with iterative design is that changes in the user interface to solve one usability problem can bring new usability problems. Therefore iterative design and evaluation should be combined [Nielsen, 1993].

6.5.2. Relevant user requirements

I-NF-1	Usability
I-NF-2	Accessibility
I-NTF-8	Look and Feel
I-NF-11	User guide is needed to assist users in system navigation and task accomplishment.

6.5.3. Design considerations

Novice vs. expert users

The users of the OCOPOMO platform will rather be novice than expert computer users.

Bandwidth

The web access for participation tools should have a simple interface that is even responsive if the bandwidth is low (e.g. no big images on the web site). Otherwise the usage of the platform could be unattractive for certain user groups.

Mobile phones

The client user interfaces to access participation tools needs not to be accessible with mobile phones.

6.5.4. Usability in OCOPOMO

Subsequently, important aspects to ensure a good usability of the OCOPOMO platform are mentioned:

Look and Feel

The OCOPOMO platform will consist of different components. In order to ensure the usability of the platform, it is necessary that the components have a consistent look and feel.

Security

The analysis of the processes and the target groups results in security requirements. Only if the users trust the system, they will be willing to use it. Hence general security requirements can be seen as user requirements. General objectives and corresponding kinds of security requirements, as e.g. identification requirements, authentication requirements etc., can e.g. be found [Firesmith, 2002]. In particular a platform like the OCOPOMO platform should be secured against sabotages to ensure the confidence of the users in the system.

Spelling

In order to make the usage easy, the spelling should be consistent in the overall platform. This includes that the same features have the same name no matter where they are on the platform (e.g. the forum should not be named forum on one site and discussions on another one). In addition, the spelling needs to be simple and self-explanatory.

Scenario

Probably, a number of end-users may exist who are not able to write a scenario on-line from scratch. Therefore there will be the need to have some core scenarios generated with facilitators -- either at real meetings or possibly virtual meetings. So, it could be an idea to offer one core scenario and the possibility for end-users easily to create branches by changing some of the assumptions (including rules)[130]. Even then, the most effective way of achieving this could be for end-users to complain about specific elements of an existing scenario or about the lack of some element and then for a moderator/facilitator to start writing and posting the alternative scenario - possibly as a result of a forum discussion. But the users need to have the opportunity to generate complete new scenarios on-line instead of letting them just participate in off-line meetings.

The dashboard

It needs to be tested in empirical testing if users can use Alfresco functionalities as e.g. customising their dashboard. The dashboard looks to be well designed for people who are seriously committed to engaging in the process. The question is if deeply involved stakeholders (our target end-users) are likely to be so committed. But as the dashboard is a standard Alfresco feature, it is included but should be tested with OCOPOMO target group. Features, which confuse the users without bringing an added value, should be removed after empirical tests.

Role specific interfaces

Often used functionalities should be provided to the users with direct links on the first webpage/start page. Different users (e.g. administrators, policy modellers, stakeholder in some simpler user role, etc.) can have a specific user interface according to their knowledge, objectives and responsibilities. This can be done with customising the dashboard for different user roles.

How-tos

Functionalities which are to be used by politicians, NGOs etc. as discussion forums and content management functionalities, need to be self-explanatory and usable without any further assistance. Simple how-tos can support this. Functionalities, which are used by policy modellers, should be described in short videos and with a manual.

[130] A branch scenario should start with a specification of the differences from the trunk scenario – see the versioning requirements.

6.5.5. Application to relevant views

Functional view

Particular usability issues are served in different way:

- Due to fact that mobile devices are not expected to be specifically used within the scope of the project (in their own specific form for views/layouts), only standard layout of the application will be available (there is no need to add user interface module for such specific views).
- How-tos (help and assistance) will be incorporated within every functionality (where applicable and per user interface of particular managers) and specific manuals/tutorial materials (expected to be available after integration workpackage).
- Security on access rights and data provision level is controlled by User manager (with all authorization aspects based on roles) combined with Process manager (for context-based views), authentication is reused within web/application server (e.g. Alfresco).
- Look and feel should be realised according to particular managers with their user interfaces and according to users' expectations described by mock-ups within the implementation of particular components (available as a part of interaction view).
- Role-specific interfaces and customised dashboard are achieved by user profiles (available using User manager) and customisation of shared space (Collaboration space manager) according to context (from Process manager), access rights and role. Additionally, dashboard customisation is available in Alfresco Share tool, therefore it is easy to reuse this feature directly in user interface of shared space.
- Scenario and spelling issues have to be in mind during the implementation of the components (especially user interfaces), but it is not needed to change current architecture.

According to previous information, there is no need for additional component(s) in the architecture to support usability issues in some specific way (all of them are served using current architecture or development process within the project).

Information View

In order to realise the requirements of the usability perspective, the existing information model has to support user profiles and access rights. The current model already has such elements. Manuals and how-tos are not specific for policy modelling information model of OCOPOMO, they will be only used as help/tutorial materials (or implemented together with particular functionalities as help inside the system). Therefore we do not need to update the information model.

7. COMPONENT FUNCTIONAL DESCRIPTION

7.1. ANNOTATION MANAGER

7.1.1. Relevant user requirements

T-39	Computer-assisted Qualitative Data Analysis Software Tool – Coding of text passages and clustering of codes
T-40	Computer-assisted Qualitative Data Analysis Software Tool – flexible querying of codes and issues
T-41	Computer-assisted Qualitative Data Analysis Software Tool – statistics
T-C1	Hints for interesting topics
NFR02_PMPM (Transformation process) - Language transition	
I-2	Transformation table – connection of context-specific information within the Scenario Generation and Policy Modelling process in ICT toolbox
I-19	Log of activities within scenario generation
New requirements:	
SOTA-6	Information structuring
SOTA-7	Memos
UC-6	Generation of relations
UC-7	Expertise-based relations
UC-8	Quantitative data analysis

7.1.2. Context of the component

A context of the Annotation Manager is depicted in Figure 26. The manager is expected to communicate with the following managers:

- Concept manager – to store and retrieve identified annotation elements (phrases, clusters, metadata)
- Content manager – to store and retrieve manager related data, providing storage functionality
- Document manager – to retrieve relevant documents (scenarios as well as other scenario-related document types)
- Link manager – to store information on relations between objects (documents and phrases, phrases and relations, relations and relations)
- Process manager – to retrieve process specific information playing the role of a context
- Search manager – to search within annotated text chunks (phrases), associated metadata and identified relations
- User manager – to obtain information on access rights and roles played by users

In addition to this, it is possible to utilise also other managers: Polling and Rating manager as well as Discussion forums manager and Chat manager for creating a communication channel linked to identified objects.

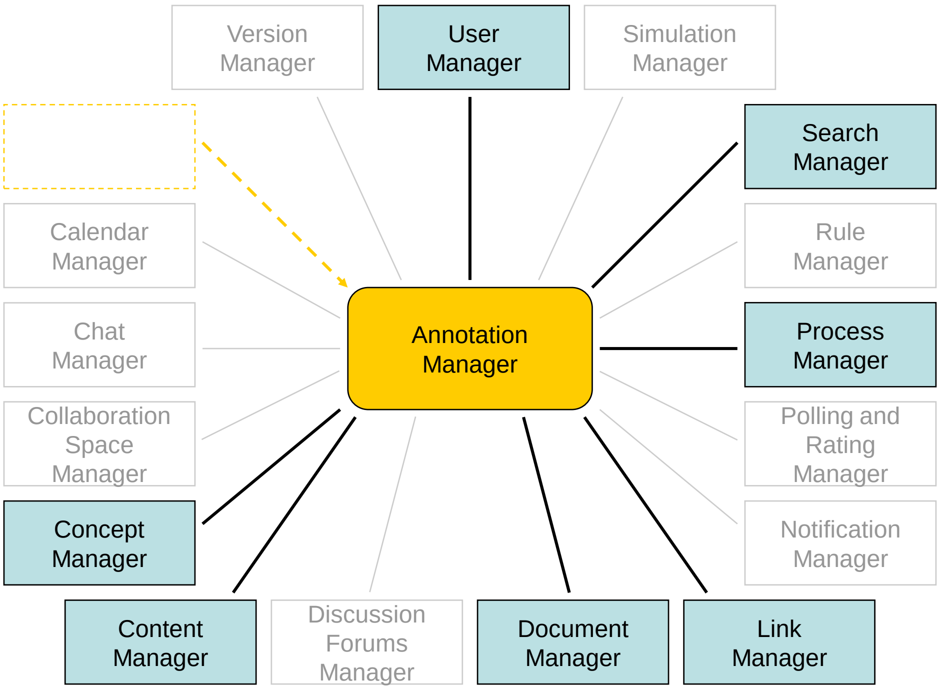


Figure 26 Context of Annotation Manager

7.1.3. Supported use cases

The component mainly provides services to users, it represents a part of the system's front-end user communicates with. Its aim is not to support other components. In the presented use case diagram, User actor represents any user-related actor that uses functionality of Annotation manager.

The manager provides users with functionality enabling to transform unstructured textual information as well as implicit expert knowledge into explicit knowledge structures. It enables to identify and select text phrases, organise them into different binary or n-ary relations, as well as to create higher order relations structuring and organising other relations. The created objects can be associated with metadata information. In addition to creation, the existing objects can be modified as well as removed from users' consideration.

All defined objects can be analysed to support users in their management. It is possible to perform qualitative analysis enabling to cluster objects/relations in order to express discovered relations among objects. To complement this, quantitative statistical analysis can be performed on identified objects as well. The objects can be compared to find differences and similarities.

Visualisation represents a window into complicated relationships among phrases and/or relations. In addition to visualising objects separately, it is possible to visualise differences between objects as well. Two types of visualisation are considered: network-based (utilising links between objects) and table-based (focusing on objects and their properties).

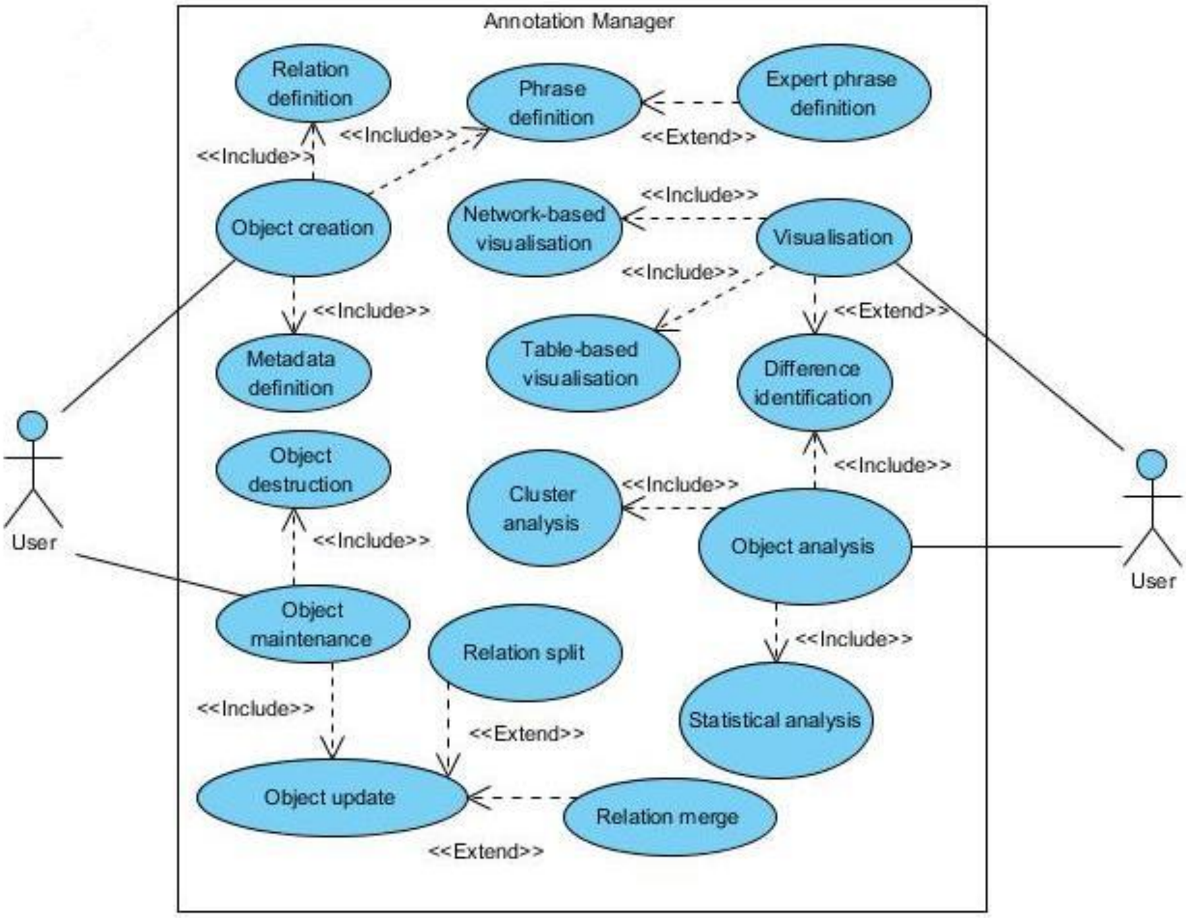


Figure 27 Use cases supported by Annotation Manager

7.1.4. Functionality description

Basically, the manager serves as a document analyser. Its aim is to put a structure upon unstructured representations and therefore transform knowledge hidden in documents into explicitly represented conceptual knowledge structures. Documents to be analysed can have two forms:

- textual documents represented explicitly as computer files utilising one of accepted document formats,
- mental documents representing expert knowledge (i.e. implicitly represented documents located in experts' minds).

In order to identify relevant information in textual documents, users have the possibility to read documents and select phrases (text passages) manually. The identified phrases are coded and stored. To manage defined phrases, users can revise the selection - they can modify phrases (extend, shrink or move them across text) as well as remove them completely. To support orientation, phrases are visualised as highlighted fragments over text representation using different colours reflecting characteristics of the phrases.

Identified phrases can be processed in two ways, creating binary or n-ary relational knowledge structures:

- relations between phrases can be defined by selecting two phrases and an appropriate dependency type (it is possible to relate phrases within one textual document as well as phrases form different textual documents)
- phrases can be clustered by selecting phrases (from one or more textual documents) and naming the cluster appropriately

Management of the created relational structures is available - the structures can be modified and/or deleted. Moreover, phrase clusters can be split into more clusters or several clusters can be merged into one cluster.

To deal with experts' mental documents, experts can define phrases as well (since their mental documents are not explicitly represented, phrases cannot be selected but must be typed). It is possible to process these phrases in the same way as the phrases identified in textual documents (both phrase types can be mixed). In addition, experts can define phrase clusters without the necessity to define and code phrases (so called expertise-based clusters).

To support identification of different relationships, the manager enables to define and manage relations over already existing binary as well as n-ary relations, for example it is possible to cluster relations between phrase pairs or relate different phrase clusters. To support creation of these higher level relations, the manager provides functionality to analyse existent relations (e.g. if two phrases, belonging to different clusters, are related, then the given clusters may be related as well), comparing them (e.g. if two clusters share one or more phrases then they may be related) and visualising the relations (either as network of relations or using a table-based format).

All defined elements (text phrases and relations) can have assigned metadata. Some metadata are expected to be filled in an automatic way (e.g. author or context), the others are user defined. Full-fledged management facilities for metadata are provided to users. In order to support different views and/or different evolution stages, all elements can be versioned (different versions of an object can exist) and versions of the same object can be compared to indicate differences.

All elements, the component manipulates with (phrases, relations and metarelations), are expected to be interlinked in order to preserve the 'cause-result' relationship. These links enable visualisation of the dependence among different objects, enabling going back and forth in this network (e.g. going from a document to a phrase, from a phrase to a cluster, from a cluster to a higher order relation, etc.).

In order to enable quantitative analysis as well, the manager provides statistical support on different levels (from words through phrase clusters to higher order relations).

Communication with Search manager is actually proposed in the way of a direct search within content repository elements related to Annotation manager. When needed, API can be added in the revision of the component.

7.1.5. Component API

none[131] (the manager does not provide functionality for other managers)

[131] Only functionality offered to other managers/modules is presented. Functionality consumed by users directly is not present in the API sections of particular managers.

7.2. CALENDAR MANAGER

7.2.1. Relevant user requirements

T-28	Shared calendar with events related to the current processes
T-42	Tags

7.2.2. Context of the component

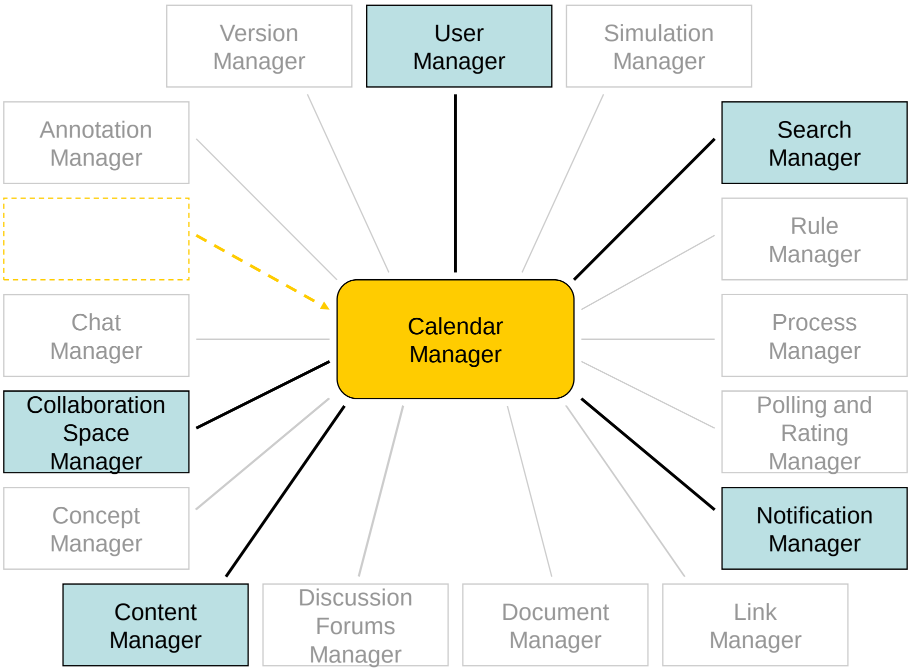


Figure 28 Context of Calendar Manager

A context of the Calendar Manager is depicted in Figure 28. This manager is expected to communicate with the following managers:

- Collaboration space manager – to provide shared calendar as a part of the collaboration space within the current project (process)
- Content manager – to store and retrieve calendar-related data to/from the content repository (especially metadata about events)
- Notification manager – to send calendar reminders through notification channel (email)
- Search manager – to provide search in calendar events
- User manager – to obtain information on access rights and roles played by users

7.2.3. Supported use cases

The component mainly provides services to users, it represents a part of the system's front-end user communicates with. Its aim is not to support other components, only use some of their APIs. In the presented use case diagram, User actor represents any user-related actor that uses functionality of Calendar manager.

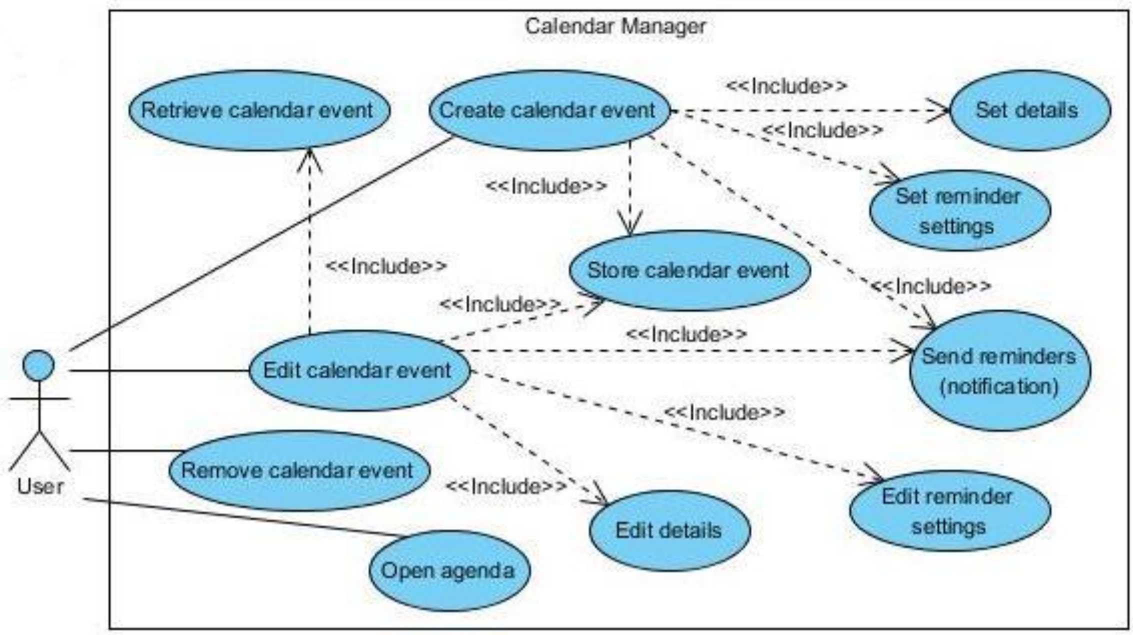


Figure 29 Use cases supported by Calendar Manager

The component represents user interface for management and visualisation of shared calendar events (and agenda). User is able to open shared calendar within the collaboration space and view existing calendar events and agenda. He/she is also able to create new calendar events (with setup of reminder function according to this event) as well as modify (edit) or remove them. As it is shared calendar, it is possible also to setup subset of user for whom the current event is important (and they will see them) or modify target users of already added calendar events.

7.2.4. Functionality description

The component serves as a user interface for calendar functionality within collaboration space of a current project (process), which includes:

- creation of new shared calendar event - it is necessary to insert date/time settings, target group of users for sharing, reminder settings, metadata information
- edit calendar event - user is able to open a calendar event and change different parts of the calendar event settings like date/time, sharing details (target users), reminder settings, metadata information
- remove calendar event
- view (open) an agenda list (a compiled list of active events)

All actions are also controlled by the access rights from the user management component. When reminder functionality is setup, system will automatically (using notification component) notify targeted users (in case of calendar events usually by email). For storage and retrieving calendar events system reuses calendar tool functionality and/or content manager for (automatic) storage of calendar event and its details/metadata. Search is done on the side of Search manager, which can directly search in data related to calendar within content repository. Context is (intentionally) shared using collaboration space and its personalisation/customisation to current process.

7.2.5. Component API

none (the manager does not provide functionality for other managers)

7.3. CHAT MANAGER

7.3.1. Relevant user requirements

T-4	Chat

7.3.2. Context of the component

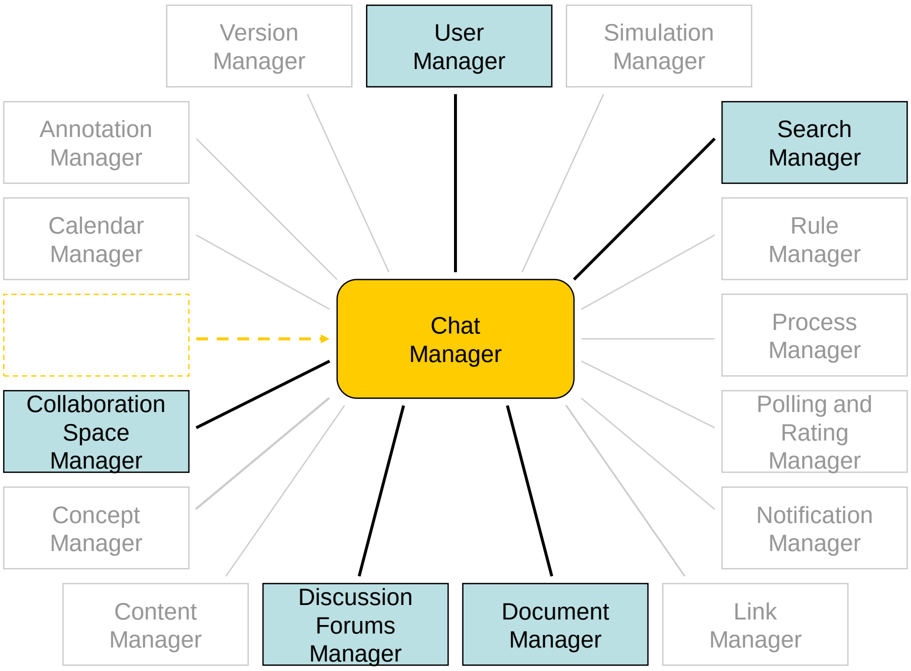


Figure 30 Context of Chat Manager

A context of the Chat Manager is depicted in Figure 30. This manager is expected to communicate with the following managers:

- Collaboration space manager – to provide chat as a part of the collaboration space within the current project (process)
- Discussion forums manager – to reuse discussion forums functionality in order to create new forum after finished chat session (if needed)
- Document manager – to store finished chat in document library as document (with metadata)
- Search manager – to provide search for text in current or finished chat(s)
- User manager – to obtain information on access rights and roles played by users

7.3.3. Supported use cases

The component mainly provides services to users, it represents a part of the system's front-end user communicates with. Its aim is not to support other components, only use some of their APIs. In the presented use case diagram, User actor represents any user-related actor that uses functionality of Chat manager.

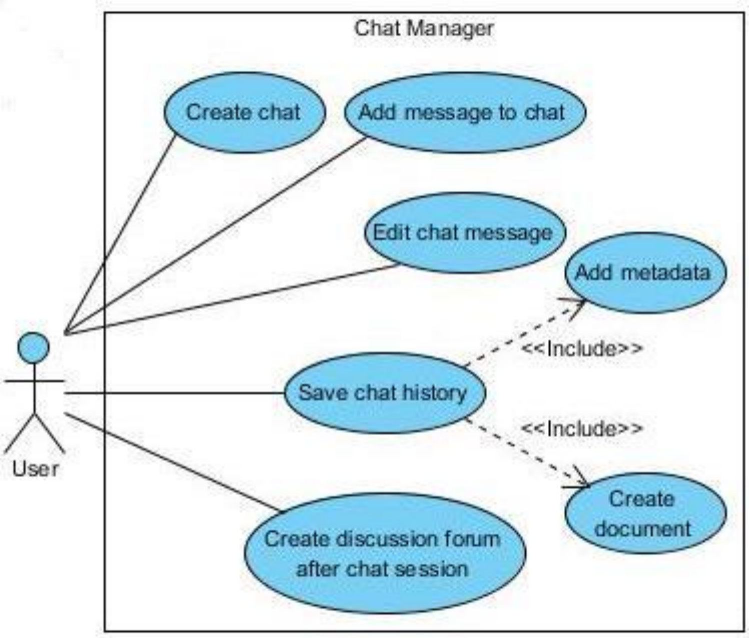


Figure 31 Use cases supported by Chat Manager

The component represents user interface for creation of simple chat room(s). Users are able to add messages to chat and communicate using simple text inputs. Users are also able to edit their messages in active chats (in order to avoid/repair typos).

Another supported action is possibility to create discussion forums after the finishing chat session. It is also possible to reuse document management functionality and create a document version of finished chat (with metadata).

7.3.4. Functionality description

The component serves as a user interface for simple chat functionality within collaboration space of a current project (process), which includes:

- creation of new chat that should have some title (topic)
- adding new messages into chat - it should be possible to react on some previous message
- storage of finished chat in one document within content management system - chat history can be saved into one document (with necessary metadata) using Document manager API in order to have data in system (if users see that it is potentially relevant for storage as a source of interesting information)
- creation of discussion forums directly related to a finished chat (if users want to follow in communication using asynchronous way)

All actions are also controlled by the access rights from the user management component. Search is done on the side of Search manager, which directly searches in data related to a chat within content repository. Context is (intentionally) shared using collaboration space and its personalisation/customisation to current process.

7.3.5. Component API

none (the manager does not provide functionality for other managers)

7.4. COLLABORATION SPACE MANAGER

7.4.1. Relevant user requirements

I-1	ICT toolbox functionality provided through one portal-based interface
I-2	Transformation table – connection of context-specific information within the Scenario Generation and Policy Modelling process in ICT toolbox
I-4	Creation of stakeholder groups for the scenario generation process
I-5	Integration of components within the e-participation tools for scenario generation – data exchange / annotation
I-7	Integration of components within the e-participation tools for scenario generation – workspace
I-23	Creation of stakeholders groups for policy modelling process
I-32	Workflow support
I-F-I5	User profile
I-F-I6	Personalise overview
New requirements:	
UC-1	Rights management
UC-4	Initiate project

UC-5 Update description of the project

7.4.2. Context of the component

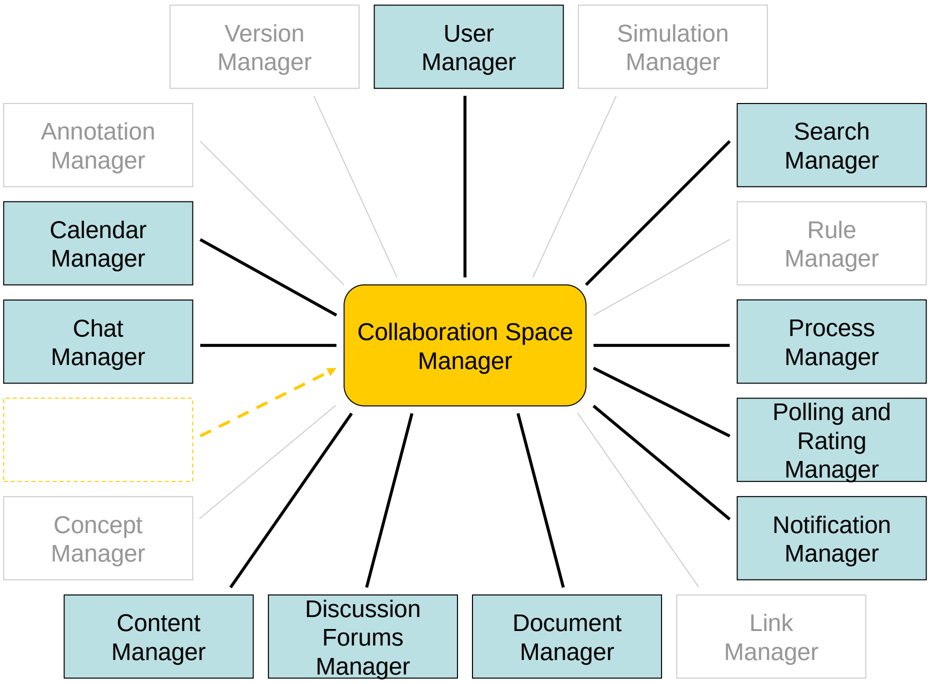


Figure 32 Context of Collaboration Space Manager

A context of the Collaboration Space Manager is depicted in Figure 32. This manager is expected to communicate with the following managers:

- Calendar manager – to manage/reuse shared calendar functionality available within collaboration space of current project (subpart: sharing of communication utilities)
- Content manager – to store and retrieve information related to collaboration space management (current project information, members, etc.)
- Chat manager – to manage/reuse chat functionality available within collaboration space of current project (subpart: sharing of communication utilities)
- Document manager – to store, retrieve and provide documents and necessary functionality from the content management part of the system
- Discussion forums manager – to manage/reuse discussion forums functionality available within collaboration space of current project (subpart: sharing of communication utilities)
- Notification manager – to publish/notify users about new artefacts and elements in space through different channels (email, news, newsletter, RSS, hints, etc.)
- Polling and Rating manager – to manage/reuse polling functionality and rating (feedback-based and tagging) of elements within collaboration space of current project (subpart: sharing of communication utilities)
- Process manager – to retrieve process specific information playing the role of a context
- Search manager – to provide search for objects in collaboration space in federated view (combination of different resources available in space)

- User manager – to obtain information on access rights and roles played by users, and also to provide profile preferences

7.4.3. Supported use cases

The component provides some services to users, but mainly serves as shared space for different tools (discussion forums, chat, calendar, document library, etc.) and some specific services for customised and personalised overview of current process. In the presented use case diagram, User actor represents any user-related (and authorised for this action) actor that uses functionality of Collaboration space manager. Application actor represents generally any tool shared within the space.

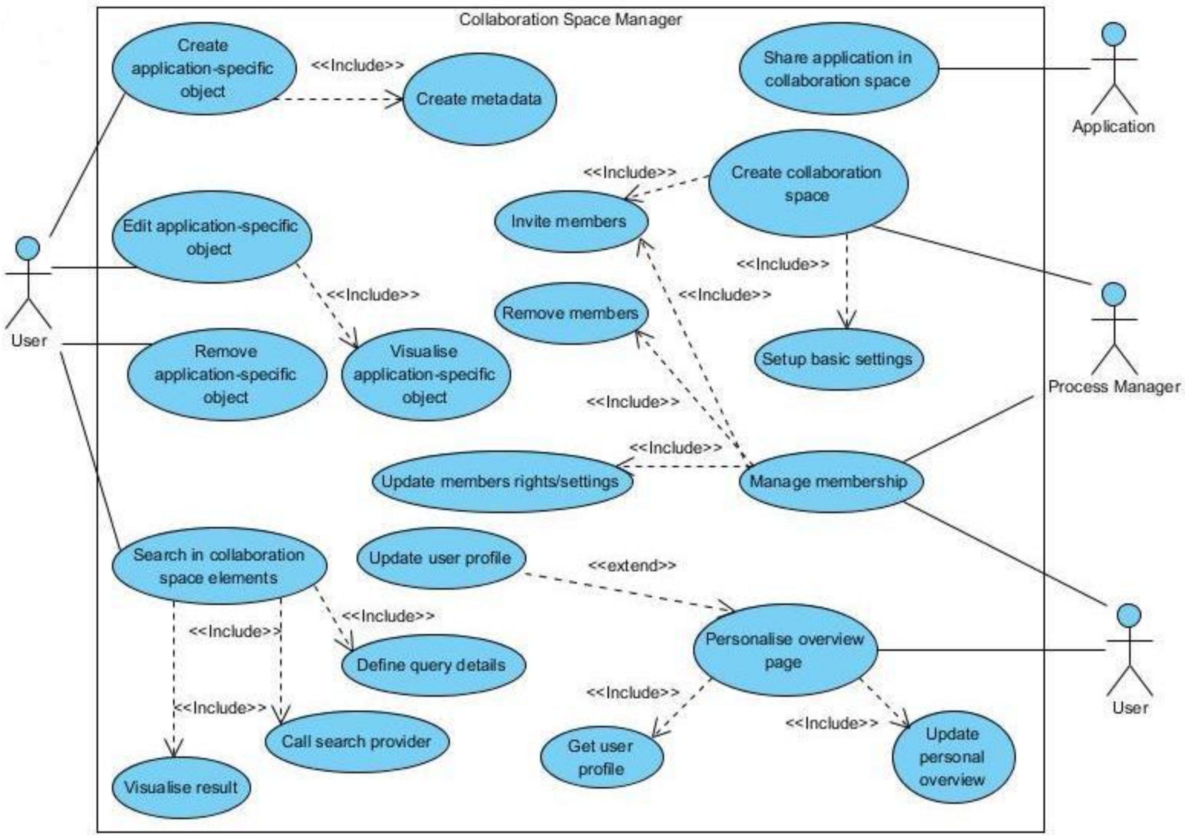


Figure 33 Use cases supported by Collaboration Space Manager

The component basically represents a collaboration space for users and applications to provide shared space together with some management and personalisation features. The presented use cases indicate processing and manipulations with collaboration space elements, which can be seen as a ‘container’. Collaboration space manager also supports creation of such shared spaces to specific process(es).

It is possible to create, use (visualise), update and destroy application specific objects (e.g. discussion forums elements, chat, calendar events, document library elements, hints, RSS readers, news, newsletters, etc.) by reusing their particular APIs. Therefore, collaboration space provides general place for sharing its ‘space’.

Other functionality focuses on the federated (combined) search through different data/information types (application elements), management (administration of current collaboration space, members management, etc. – all included within ‘Manage membership’ use case) and customisation or personalisation of collaboration space (setup of personal start page for group and for particular user, users’ ability to personalise their customised pages).

7.4.4. Functionality description

The component serves as a ‘container’ for sharing different data and applications (tools) in one shared space with personalisation features. Its aim is:

- to create shared space in order to support initiation of project (process)
- to provide ‘space’ (in general) in collaboration space for sharing applications (tools) and its data (if sharing is needed for accomplishing particular tasks)
- to manage collaboration space settings (administration of shared space, members management, etc.)
- to personalise start page and its ‘attention management’ features
- to provide results related to federated search in different applications information and data types (data models)

Creation of collaboration space is provided directly by this component and should be used by other components. In our architecture it is Process manager that helps in initiation of the projects (process of policy modelling). It includes specification of project details and basic management of created space (basic access rights setting, invitation of users – starting group, creation of and access to basic documents). Management of collaboration space is then (after creation) also available to authorized users directly within the shared space interface.

In general, it is expected that collaboration space provides (as simple as possible) support for involvement of any tool suitable for incorporation into shared space (within specific dashlet/portlet or some specific tab). This can be achieved mostly by implementing some specific class or user interface characteristic for technological framework under the OCOPOMO platform.

Personalisation and customisation of the shared space and user’s personal overview pages is done within collaboration space and its manager supports it as a functionality for users. While shared space can only be customised by administrators of such space, personal overview pages and specific (profiled) settings on tools are managed by all involved users individually according to their user profiles.

Shared space combines many different types of data and applications. Therefore user is able to reuse search component for providing combined/federated searches through different parts of the OCOPOMO data model. Search specific to collaboration space members is done on the side of Search manager, which directly searches in data related to shared space memberships within content repository.

7.4.5. Component API

Function	Description
Create collaboration space	*To initiate process-specific shared space.* **Input**: process – an identification of a process instance (context) details – basic information regarding start of the process, e.g. textual details, basic settings, administrator, basic set of involved users **Output**: space – new collaboration space with process-specific context
Support tool in collaboration space	*To provide functionality/interface within the collaboration space for particular tool (register application).* **Input**: tool – an identification of a tool (application) for incorporating to collaboration space space – an identification of targeted collaboration space **Output**: result type – to identify whether registration was successful or failed

Table 52 Collaboration Space Manager API

7.5. CONCEPT MANAGER

7.5.1. Relevant user requirements

T-23	PM (Analysis) - Qualitative representation of the simulation results
T-39	Computer-assisted Qualitative Data Analysis Software Tool – Coding of text passages and clustering of codes
T-40	Computer-assisted Qualitative Data Analysis Software Tool – flexible querying of codes and issues
T-41	Computer-assisted Qualitative Data Analysis Software Tool – statistics
FR01_PM	PM (Transformation process) - Define initial policy modelling aspects
FR02_PM	PM (Transformation process) - Stakeholder extraction
FR03_PM	PM (Transformation process) - Environment generation
FR04_PM	PM (Transformation process) - Goal definition
FR05_PM	PM (Transformation process) - Rule generation
FR06_PM	PM (Transformation process) - Assumption definition
FR07_PM	PM (Modelling process) - Agent type creation
FR08_PM	PM (Modelling process) - Agents at different aggregation levels
FR09_PM	PM (Modelling process) - Exogenous factors
FR10_PM	PM (Modelling process) - Environment definition - general
TP-1	PM (Analysis) – Within-timestep dependency graph visualisation
TP-2	PM (Analysis) – Experiment and rule development browser
TP-3	PM (Analysis) - Narrative output

NFR01	PM (Transformation process) - Data representation
I-2	Transformation table – connection of context-specific information within the Scenario Generation and Policy Modelling process in ICT toolbox
I-5	Integration of components within the e-participation tools for scenario generation – data exchange / annotation
I-12	Support for direct export/import of information between scenario generation process and policy modelling
I-14	Maintaining of scenarios and rules within the ICT toolbox
I-25	Integration of policy modelling tool and simulation / analysis tools – data exchange / annotation
I-40	Transition table browser

7.5.2. Context of the component

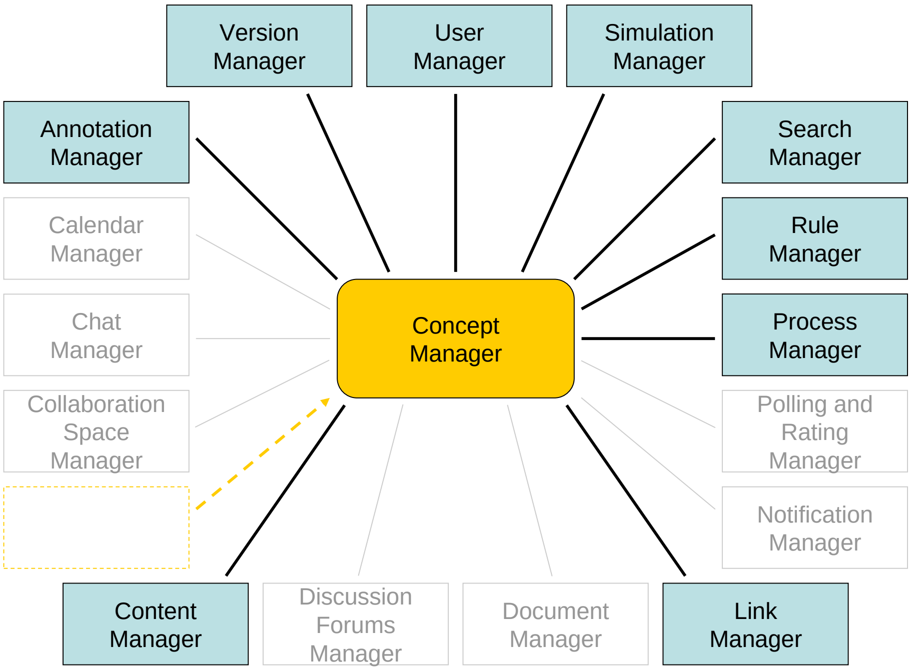


Figure 34 Context of Concept Manager

A context of the Concept Manager is depicted in Figure 34. This manager is expected to communicate with the following managers:

- Annotation manager – to provide functionality for creation of conceptual description elements
- Content manager – to store and retrieve concept to/from the content repository
- Link manager – to use link functionality for knowledge modelling and connecting concepts with other data/knowledge sources and elements
- Process manager – to retrieve process specific information playing the role of a context
- Rule manager – to provide concepts from conceptual description in definition and evaluation of rules and agents

- Simulation manager – to provide concepts from conceptual description in definition and evaluation of simulation models and simulations
- Search manager – to provide search in concept objects and its metadata
- User manager – to obtain information on access rights and roles played by users
- Version manager – to support versioning in storage and retrieving of conceptual descriptions (CCD)

7.5.3. Supported use cases

The component provides service for creation and management of conceptual description objects to particular managers in order to support them in knowledge-based policy modelling process and evidence-based description of problem.

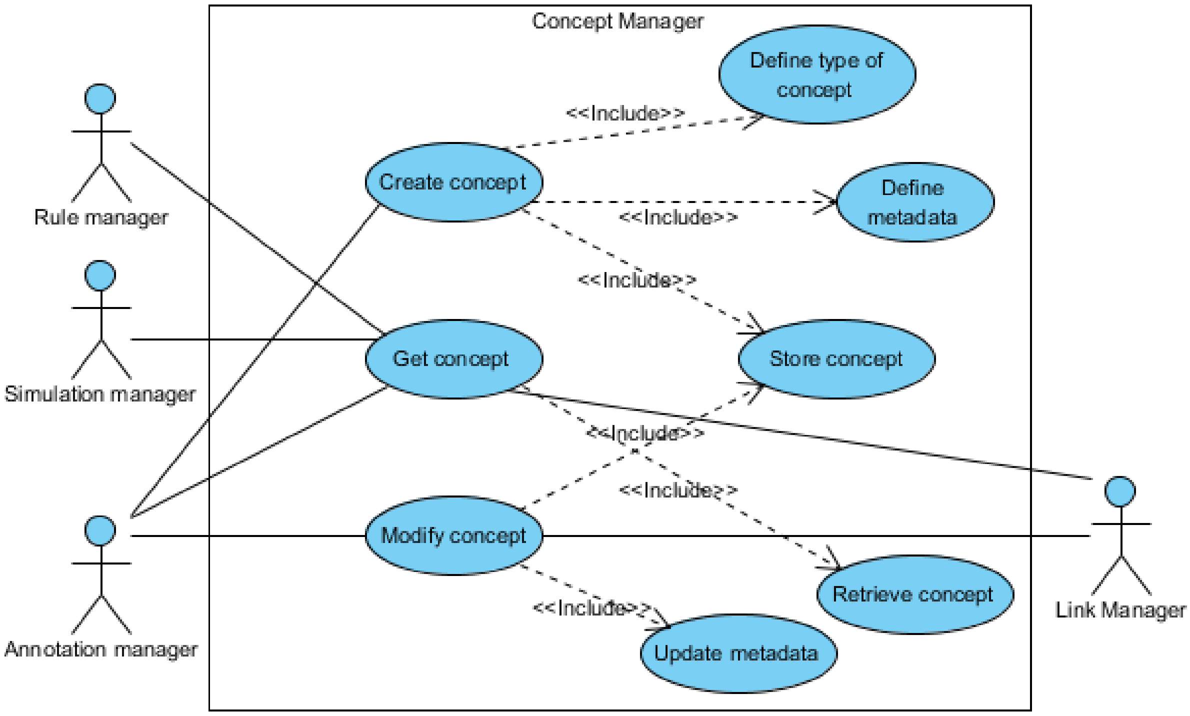


Figure 35 Use cases supported by Concept Manager

The component is able to work with concepts from conceptual descriptions – create, modify and remove them from the structured model of CCD. The presented use cases show processing and manipulations with such objects. In general, component provides services for other managers (Annotation, Rule, etc.) to create, modify, remove and access concepts on the one side (with its specific metadata), and to use content repository in order to store and retrieve concepts on the second side (low ‘data’ level of concepts definitions).

Visualisation of concepts is expected to be provided by other components (Rule manager, Annotation manager, etc.). Links between CCD elements (concepts) and other parts of data model are modelled using different manager (Link manager). Concepts can be versioned using versioning support of other manager (Version manager) during the process of storing the concept.

7.5.4. Functionality description

The component serves as a ‘middleware’ between managers for direct (user-based) support of scenario analysis and policy modelling process and low-level repository storage, where conceptual descriptions are created and managed. For example, Rule manager is using already created concepts (from Annotation manager) and adds links to them, but also is able to create another concepts and new links in order to create sufficient knowledge base for creation of a simulation model. So the main objectives are:

- to provide methods for creation of concepts (conceptual description elements) of different types (structured information with part of rules, agents, actors, networks of them, etc.) with metadata definition according to identified data model
- to provide concepts (on demand) for other components
- to support persistence of conceptual descriptions within content repository – storage and retrieval of concepts and their metadata
- to support versioning of concepts within repository (using Version manager)

More details regarding conceptual definition of objects is written within specialised components for scenario analysis and policy modelling (Annotation manager, Rule manager, Simulation manager). In general, concepts are a part of the linked and structured information from evidence-based data to simulation models with agents and rules. Search is done on the side of Search manager, which directly searches in data related to concepts within content repository.

7.5.5. Component API

Function	Description
Create concept	*To create conceptual description element of defined type.* **Input**: type of concept – a definition of type concept element data object – an object with all data necessary to create defined type of concept and its metadata **Output**: concept – an identification of created conceptual description element
Modify concept	*To modify concept definition and data.* **Input**: concept – an identification of a concept data object – an object with necessary data and definitions to modify specified concept and/or its metadata **Output**: result type – to identify whether update of concept was successful or failed
Get concept	*To obtain concept from conceptual description.* **Input**: concept – an identification of a concept **Output**: concept object – retrieved object of specified conceptual description element

Table 53 Concept Manager API

7.6. CONTENT MANAGER

7.6.1. Relevant user requirements

T-5	Content Management System (CMS) functionality
I-25	Integration of policy modelling tool and simulation / analysis tools – data exchange / annotation

7.6.2. Context of the component

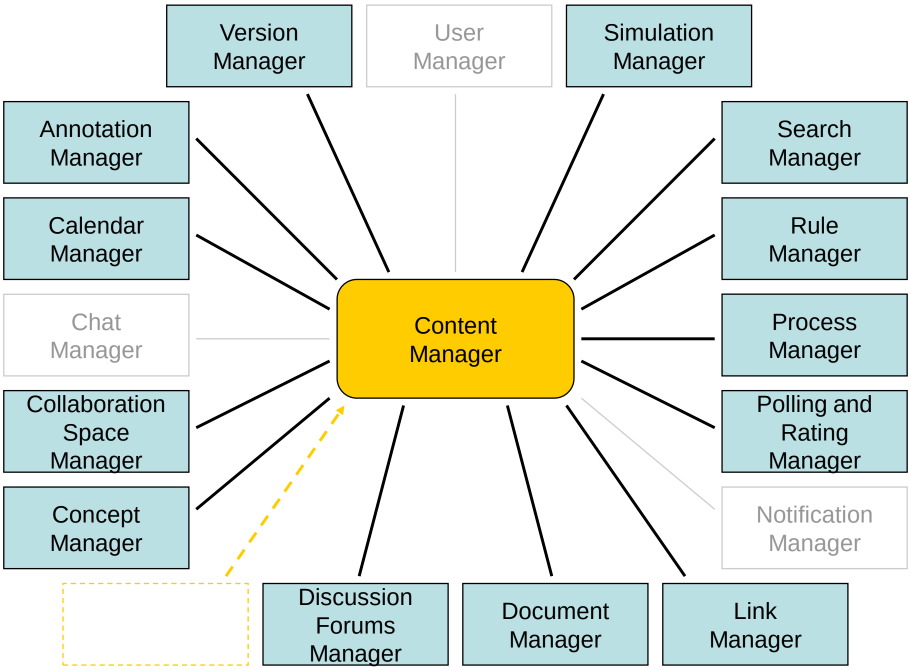


Figure 36 Context of Content Manager

A context of the Content Manager is depicted in Figure 36. This manager is expected to communicate with the following managers:

- Annotation manager – to store/retrieve manager-specific data (annotations and its combinations, evidence-based data, etc.)
- Calendar manager – to store/retrieve manager-specific data (calendar events and its metadata)
- Collaboration space manager – to store/retrieve manager-specific data (shared space settings, memberships of users, etc.)
- Concept manager – to store/retrieve manager-specific data (concept objects)
- Discussion forums manager – to store/retrieve manager-specific data (discussion forums and their elements with metadata)

- Document manager – to store/retrieve manager-specific data (documents and types of documents, all with metadata)
- Link manager – to store/retrieve manager-specific data (link objects)
- Polling and Rating manager – to store/retrieve manager-specific data (polls and relevance feedback data)
- Process manager – to store/retrieve manager-specific data (context)
- Rule manager – to store/retrieve manager-specific data (rules, agents, etc.)
- Search manager – to support search in different content repository objects and store/retrieve manager-specific data
- Simulation manager – to store/retrieve manager-specific data (simulation models, results, etc.)
- Version manager – to support versioning of content repository objects

7.6.3. Supported use cases

The component provides services to other components for storage and retrieval of objects to/from content repository. In provided use case diagram Manager actor represents any manager that uses services of Content manager to store/retrieve content repository object. Additionally, search for specific content repository objects (and their metadata) is provided to Search manager.

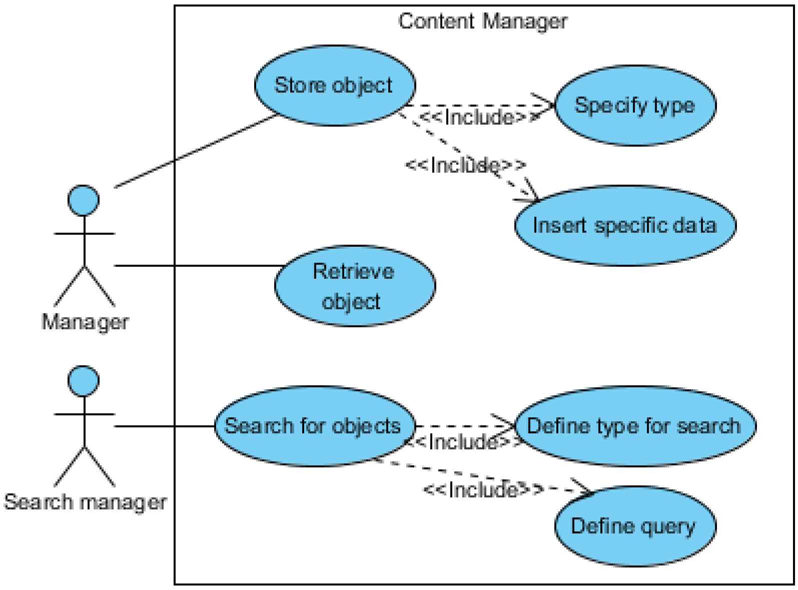


Figure 37 Use cases supported by Content Manager

The component represents a service for content repository management. Instead of Document manager, where user interface to document management is important, here we have more low-level vision of different types of objects and functionality for their persistence and accessing by other components (provided by them to users).

Main use cases are store (also used as modifying) and retrieve objects to/from content repository. Combination of them is used also for modifying, but this is done on the side of particular components (they retrieve object, modify it and then store back in repository, with versioning support, if needed).

Search manager has (additionally to standard store/retrieve functions) search in content repository objects and retrieve them as query results.

7.6.4. Functionality description

The component serves only as a service for other components in order to store/retrieve functionality for them in content repository. So the main objectives are:

- to store object – component specifies type of an object and inserts data into object, which is then stored
- to retrieve object – object is retrieved according to its identification
- to search in objects – Search manager specifies a query and types of objects for search, relevant objects are then returned as a result set (hits)
- to support versioning of objects in content repository (using versioning component)

Context-specific information (if it is needed) is obtained using service of Process manager.

7.6.5. Component API

Function	Description
Store object	*To store object of defined type in content repository*. **Input**: type – data-model definition of type of object for storage data – object of such type with specific data (if object is going to be modified, data have existing ID, otherwise it is generated) **Output**: object – an identification of currently stored object in repository
Retrieve object	*To retrieve object from content repository*. **Input**: identification – unique identificator of object to be retrieved from the repository **Output**: object – retrieved object from the repository
Search for objects	*To provide search in content repository for defined type of objects and according to specified query*. **Input**: type – defined type of objects to be searched query – specified query to repository **Output**: hits – set of objects returned as results to specified query

Table 54 Content Manager API

7.7. DISCUSSION FORUMS MANAGER

7.7.1. Relevant user requirements

T-1	Discussion forums
T-1-1	Discussion forums - multiple instances of a forum
T-1-2	Discussion forums - entries should be organised in threads
T-1-3	Discussion forums - possibility to order entries in chronological order and for topics
T-1-4	Discussion forums - Authorisation on level of the discussion forum
T-1-5	Discussion forums - condition of use
T-12	Discussion forums – moderated and non-moderated discussions
T-14	Discussion forums – rating of contributions and contributors (analysis of discussions based on a relevance feedback)
T-42	Tags
I-5	Integration of components within the e-participation tools for scenario generation – data exchange / annotation
I-17	Discussion about simulation results and decisions of human agents in simulation

7.7.2. Context of the component

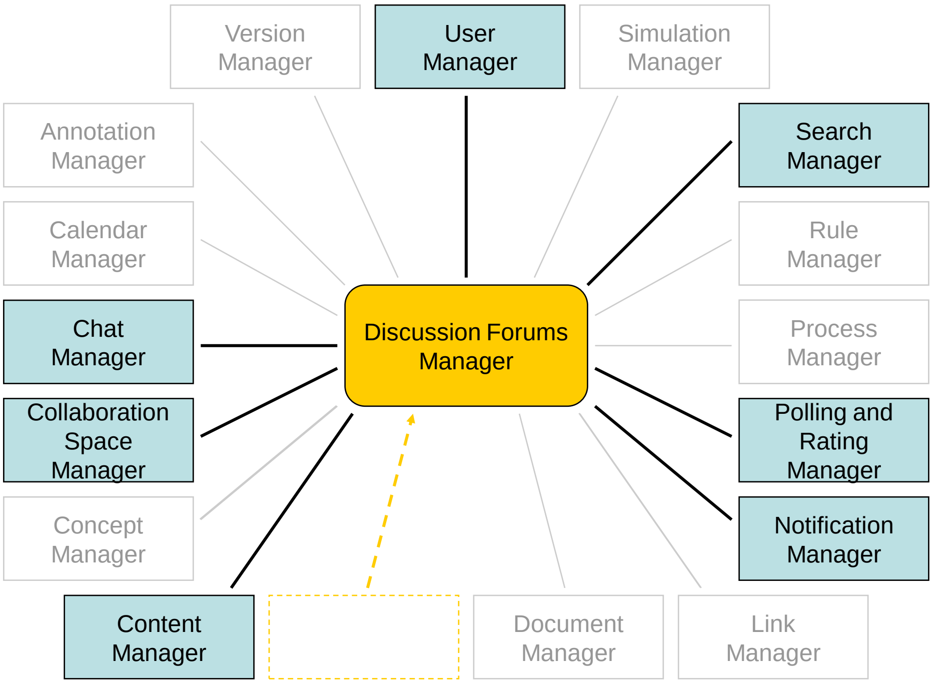


Figure 38 Context of Discussion Forums Manager

A context of the Discussion Forums Manager is depicted in Figure 38. This manager is expected to communicate with the following managers:

- Collaboration space manager – to provide discussions as a part of the collaboration space within the current project (process as a context)
- Search manager – provides search functionality through discussion forums/threads
- Notification manager – notifies users about new messages in relevant discussion forums/threads
- User manager – ensures access authorisation to a particular forum/thread
- Polling and rating manager – provides possibility to poll on facts mentioned in a discussion forum thread/message and/or rate messages in threads
- Content manager – to store and retrieve manager specific data
- Chat manager – to provide a discussion thread/message to finished chat

In addition to this, it is possible to utilise also Polling and Rating manager for creating a rating channel linked to (a set of) discussion thread messages.

7.7.3. Supported use cases

Aim of the component is to provide discussion forum functionality to other relevant components as well as to users. Different types of actors are provided, where User is most general one.

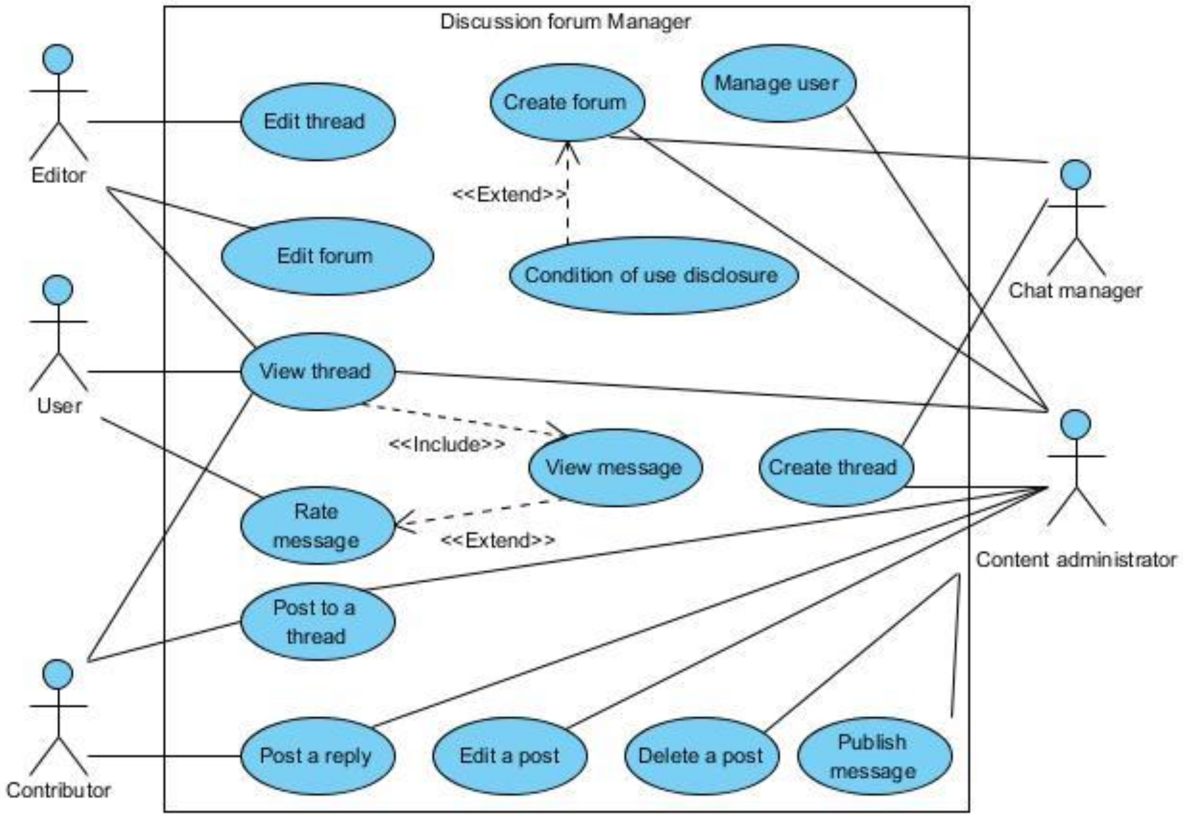


Figure 39 Use cases supported by Discussion Forums Manager

The component represents a relatively independent module, which provides presented functions related to discussion forums – to create forum/topic, to post messages and to publish, edit and delete messages for moderated forums.

Discussion forum/discussion thread can be created for and linked with all relevant objects created in the collaboration space component. Messages/entries can be organised in different types of order (e.g. chronological, topical).

7.7.4. Functionality description

The purpose of the component is to provide a possibility for users to discuss various topics related to policy development. First, an authorised user has to create a forum and after that a required number of discussion threads can be created. Threads are linked with relevant objects to give participants possibility to discuss relevant topics. For each forum/thread authorized users have to be defined.

In addition to it, the discussion forum manager provides some other managers with the possibility to create forums/threads attached to artefacts these managers deal with (e.g. documents, knowledge structures like social nets or agents, rules and rule/data dependency graphs, simulation outcome, etc.).

Each discussion forum and thread has its description, which is entered during forum/thread creation and maintained later by an editor.

Authorized contributors can send messages to a forum/thread. For non-moderated forum, a message, which was sent, is displayed immediately after it is received by the module. For moderated forums/threads a message is waiting in the queue until it is approved by a moderator. Content administrator has a right to edit and/or delete messages.

Users are able to attach a relevance feedback to contributions in discussion forums using a rating scale (e.g. 2 – strongly agree, 1 – agree, 0 – neutral, -1 – disagree, -2 – strongly disagree) about the content. After finishing a chat, responsible user is able to create discussion forum for following discussion in the asynchronous way. Search is done on the side of Search manager, which directly searches in data related to discussions within content repository.

7.7.5. Component API

Function	Description
Create discussion forum	*To create a new discussion forum* **Input**: Identification of forum (forum name) **Output**: discussion forum "handler"
Create discussion thread	*To create a new discussion forum thread* **Input**: Identification of thread (thread name), identification of the relevant discussion forum **Output**: discussion forum thread "handler"
Create message	*To create a new thread message* **Input**: Identification of relevant discussion forum and discussion thread as well as parent message (to respond to some specific message)

	Output: message “handler”

Table 55 Discussion Forums Manager API

7.8. DOCUMENT MANAGER

7.8.1. Relevant user requirements

T-5	Content Management System (CMS) functionality
T-39	Computer-assisted Qualitative Data Analysis Software Tool – Coding of text passages and clustering of codes
T-42	Tags
T-C1	Hints for interesting topics
I-24	Publishing of simulation results by the publishing tool (content management tool)
New requirements:	
SOTA-1	Workflow engine
SOTA-2	Content/WYSIWYG
SOTA-3	File types supported
SOTA-4	Several document editors
SOTA-5	Real-time co-editing
SOTA-7	Memos
UC-5	Update description of the project

7.8.2. Context of the component

A context of the Document Manager is depicted in Figure 42. This manager is expected to communicate with the following managers:

- Annotation manager – to give possibility to annotate documents
- Chat manager – to have possibility to save chat session in the form of a document for documenting the chat session
- Collaboration space manager – to provide place, where documents will be arranged together with other ones
- Content manager – to utilize content repository functionality
- Notification manager – to use notification functionality related to document-based events
- Search manager – to provide full text search within documents combined with attribute-based as well as metadata-based search
- Simulation manager – to document simulations (simulation outputs – model-based scenarios)
- User manager – to provide document access rights verification
- Version manager – to provide versioning functionality during document development

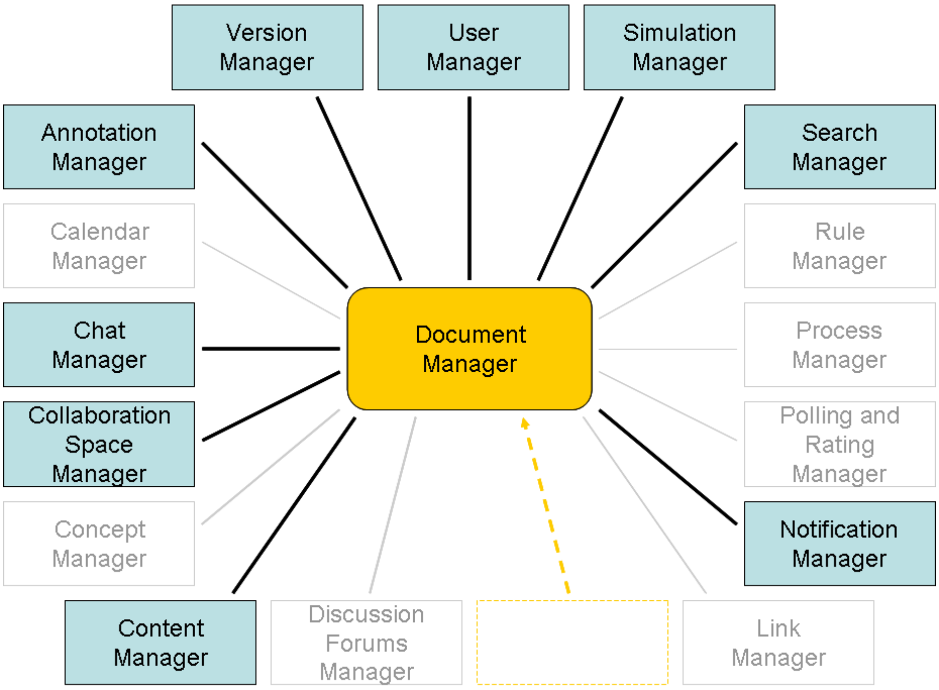


Figure 40 Context of Document Manager

7.8.3. Supported use cases

The aim of the component is to provide document management functionality to users (functionality of the manager as well as mediated functionality of other relevant components). Component represents a relatively independent module, which provides presented functions related to documents handling - to create, edit, delete, version and tag documents. In the presented use case diagram, User actor represents any user-related actor and Manager represents any manager (e.g. Chat, Simulation) that uses functionality of Document manager.

Documents are organized within a collaboration space and can be linked with all relevant objects. Presented functionality supposes the possibility to create a new document as a native document created by means of the designed system (internal editor) or by uploading documents of different formats.

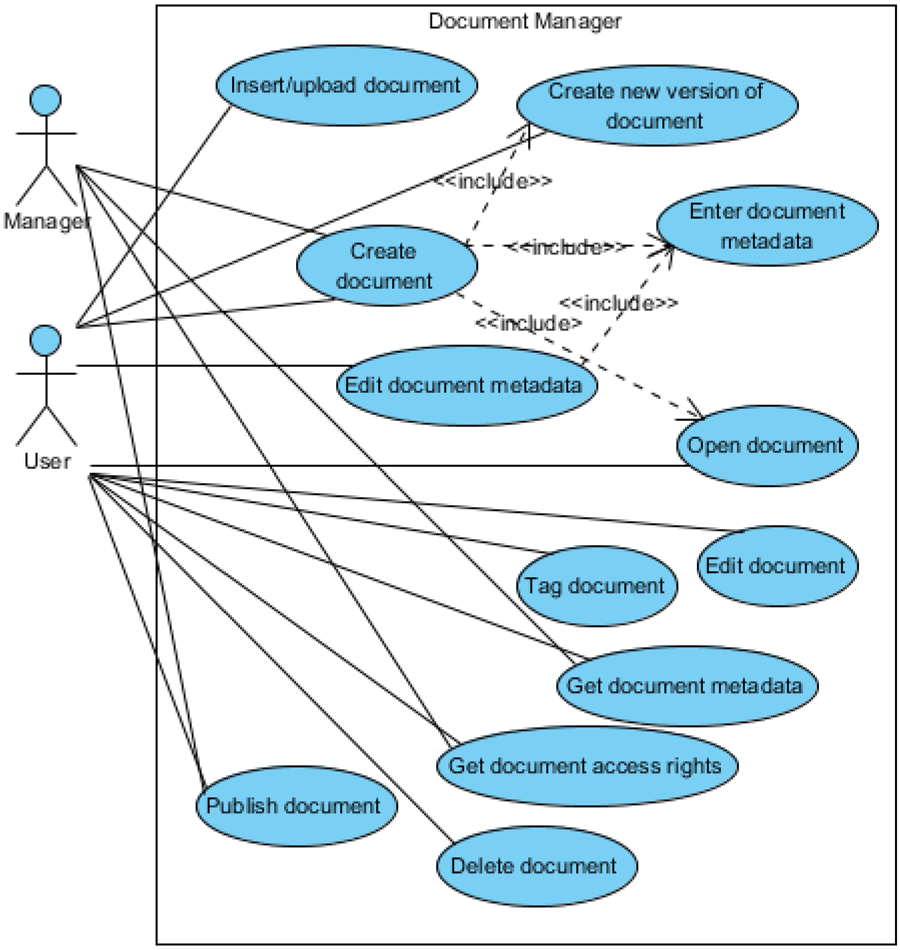


Figure 41 Use cases supported by Document Manager

7.8.4. Functionality description

The Document manager provides support for full life-cycle (workflow) of documents - creation, editing and deletion of various types of documents. Created documents are organized in one part of the collaboration space (document library) and are provided to other users after publishing them by owners (creators) of the documents.

A user is provided with the functionality to create a new document. In the case versioning is switched on, an initial version (version one) of the document is created. The document can be opened immediately after its creation or the action can be postponed. An open document can be edited and tagged – some parts of the document can be highlighted and linked to other object of the system (utilizing functionality of Annotation Manager). In the case versioning is switched off, document can be deleted from the system.

New documents can be created also by uploading various types of documents - text files (e.g. DOC, PDF, ODT, etc.), spreadsheets, presentations, etc.

Various metadata are maintained for documents. Some of the metadata are entered by user during document creation process (i.e. description of the document itself), some are maintained during document life-cycle automatically (owner, creation and access time, etc.). Manually entered metadata can be edited by an authorized user. Displaying of metadata is also provided by the component.

New document – a document just created or a new version of some existing document - is accessible for the owner of document. To provide access to the document to other users, the document has to be published by its owner.

Search is done on the side of Search manager, which directly searches in data related to documents within content repository.

7.8.5. Component API

Function	Description
Create document	*To create a new document* **Input**: description of document **Output**: document ID
Get document metadata	*To get document metadata* **Input**: document ID **Output**: document metadata
Get document access rights	*To get document's access rights* **Input**: document ID, user **Output**: access rights

Table 56 Document Manager API

7.9. LINK MANAGER

7.9.1. Relevant user requirements

T-23	PM (Analysis) - Qualitative representation of the simulation results
T-39	Computer-assisted Qualitative Data Analysis Software Tool – Coding of text passages and clustering of codes
T-40	Computer-assisted Qualitative Data Analysis Software Tool – flexible querying of codes and issues
T-41	Computer-assisted Qualitative Data Analysis Software Tool – statistics
FR01_PM	PM (Transformation process) - Define initial policy modelling aspects
FR02_PM	PM (Transformation process) - Stakeholder extraction
FR03_PM	PM (Transformation process) - Environment generation
FR04_PM	PM (Transformation process) - Goal definition
FR05_PM	PM (Transformation process) - Rule generation
FR06_PM	PM (Transformation process) - Assumption definition
FR07_PM	PM (Modelling process) - Agent type creation

FR08_PM PM (Modelling process) - Agents at different aggregation levels

FR09_PM PM (Modelling process) - Exogenous factors

FR10_PM PM (Modelling process) - Environment definition - general

TP-1 PM (Analysis) – Within-timestep dependency graph visualisation

TP-2 PM (Analysis) – Experiment and rule development browser

TP-3 PM (Analysis) - Narrative output

NFR01 PM (Transformation process) - Data representation

I-2 Transformation table – connection of context-specific information within the Scenario Generation and Policy Modelling process in ICT toolbox

I-5 Integration of components within the e-participation tools for scenario generation – data exchange / annotation

I-12 Support for direct export/import of information between scenario generation process and policy modelling

I-14 Maintaining of scenarios and rules within the ICT toolbox

I-25 Integration of policy modelling tool and simulation / analysis tools – data exchange / annotation

I-40 Transition table browser

New requirements:

UC-10 Development of social network

7.9.2. Context of the component

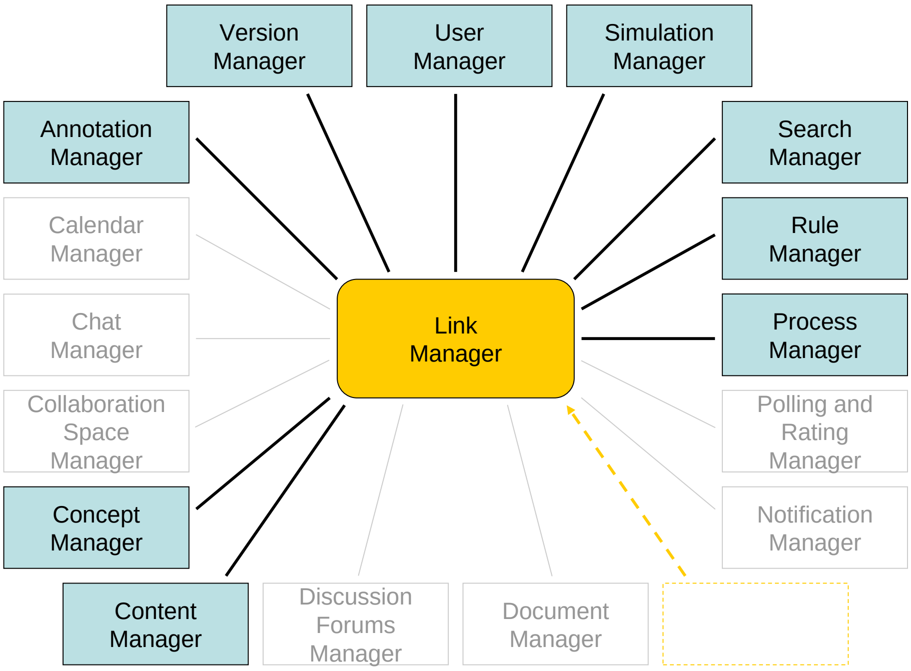


Figure 42 Context of Link Manager

A context of the Link Manager is depicted in Figure 42. This manager is expected to communicate with the following managers:

- Annotation manager – to provide functionality for creation of links between evidence-based data and conceptual description elements
- Content manager – to store and retrieve links objects to/from the content repository
- Concept manager – to create and use concepts from conceptual descriptions (CCD elements)
- Process manager – to retrieve process specific information playing the role of a context
- Rule manager – to provide links objects and functionality for definition and evaluation of rules and agents
- Simulation manager – to provide links objects and functionality for definition and evaluation of simulation models and simulations
- Search manager – to provide search in link objects and its metadata
- User manager – to obtain information on access rights and roles played by users
- Version manager – to support versioning in storage and retrieving of link objects

7.9.3. Supported use cases

The component provides service for creation and management of links between different parts of the data model within OCOPOMO scenario analysis and policy modelling processes. All this functionality is available for specific managers related to such processes. Links are helpful in evidence-based understanding of data, models and simulations.

The component is able to work with link objects – create, access, modify and remove. The presented use cases show processing and manipulations with such objects. In general, component provides services for other managers (Annotation, Rule, etc.) to create, modify, remove and access links (with their specific metadata), and use content repository to store and retrieve links (low ‘data’ level of links definitions – persistence).

Visualisation of concepts is expected to be provided by other components (Rule manager, Annotation manager, etc.). Links are connecting data sources (text annotations) with concepts in conceptual descriptions, hierarchy of concepts, actors in policy modelling process (social networks), rules, agents, and simulation models. One of the main objectives for links objects is to provide evidence-based connection of simulation results and modelling objects with real information from data sources. Concepts are modelled using different manager (Concept manager). Links can be versioned using versioning support of another manager (Version manager) during the process of link storage.

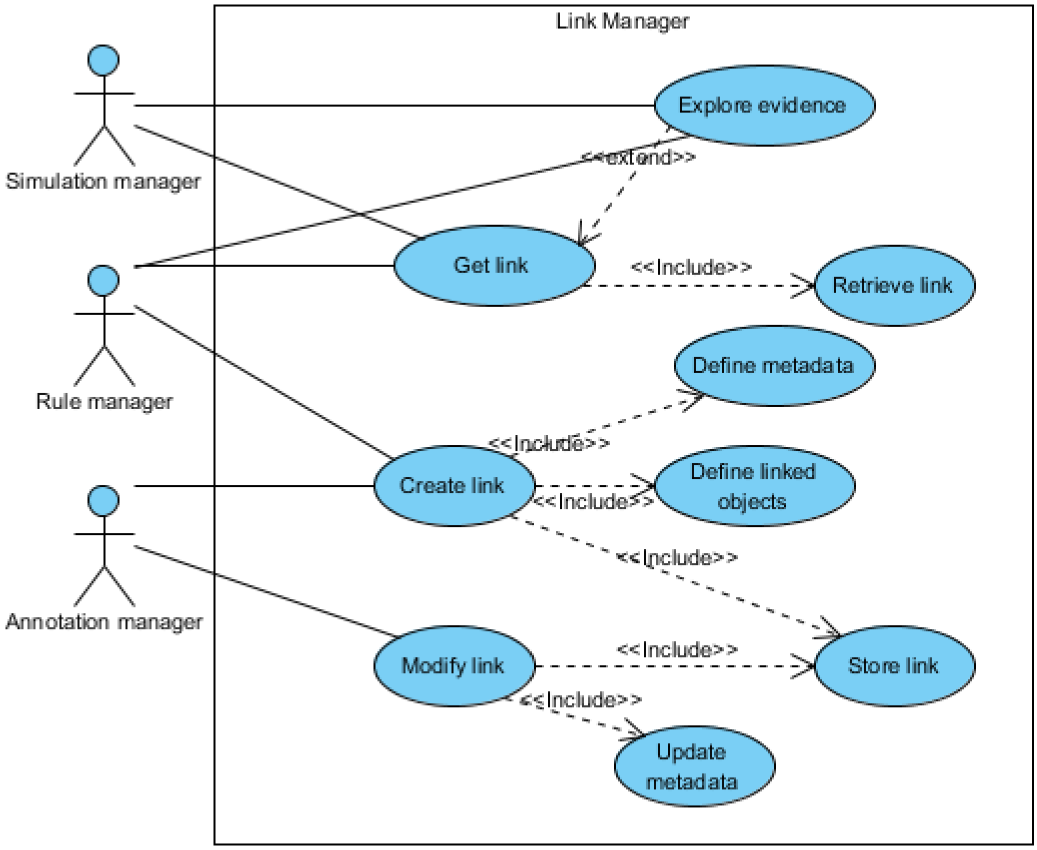


Figure 43 Use cases supported by Link Manager

7.9.4. Functionality description

The component serves as a ‘middleware’ between managers for direct (user-based) support of scenario analysis and policy modelling process and low-level repository storage, where links for knowledge-based connection of data sources and conceptual elements are created and managed. For example, Rule manager is using already created concepts (from Annotation manager) and adds links to them, but also is able to create other concepts and new links in order to create sufficient knowledge base for creation of a simulation model. So the main objectives are:

- to provide methods for creation of links between different types of structured information (data sources, text, rules, agents, actors, networks of them, etc.) with metadata definition according to identified OCOPOMO data model and data flow
- to provide links (on demand) for other components
- to support persistence of links within content repository – storage and retrieval of links and their metadata
- to support versioning of links within repository (using Version manager)

More details regarding potential links definitions is written within specialised components for scenario analysis and policy modelling (Annotation manager, Rule manager, Simulation manager). In general, links are one of the crucial parts of the data model for linked structured information from evidence-based data to simulation models (with agents and rules) where conceptual descriptions (CCD) reside in the middle. Search is done on the side of Search manager, which directly searches in data related to links within content repository.

7.9.5. Component API

Function	Description
Create link	*To create link between defined sources.* **Input**: linked object 1 – a definition of the first source for one side of link, can be of different type (data source, annotation, concept, etc.) linked object 2 – a definition of the second source for the other side of link **Output**: link – an identification of link between input objects with metadata describing link details
Modify concept	*To modify link definition and data objects.* **Input**: link – an identification of a link object change specification object – an object with structured identification of changes in link definition, data objects or metadata **Output**: result type – to identify whether update of link was successful or failed
Get concept	*To obtain link from repository storage.* **Input**: link – an identification of a link **Output**: link object – retrieved object of specified link with connection to linked objects and metadata information

Table 57 Link Manager API

7.10. NOTIFICATION MANAGER

7.10.1. Relevant user requirements

T-24	News functionality
T-28	Shared calendar with events related to the current processes
T-29	Newsletter
T-30	RSS
T-34	E-mail notification system
T-C1	Hints for interesting topics
I-4	Creation of stakeholder groups for the scenario generation process
I-23	Creation of stakeholders groups for policy modelling process
I-7	Integration of components within the e-participation tools for scenario generation – workspace
I-32	Workflow support

I-F-15	User profile
I-F-I6	Personalise overview
I-NF-11	Help and assistance

New requirements:

UC-2	Invitation – send and receive

7.10.2. Context of the component

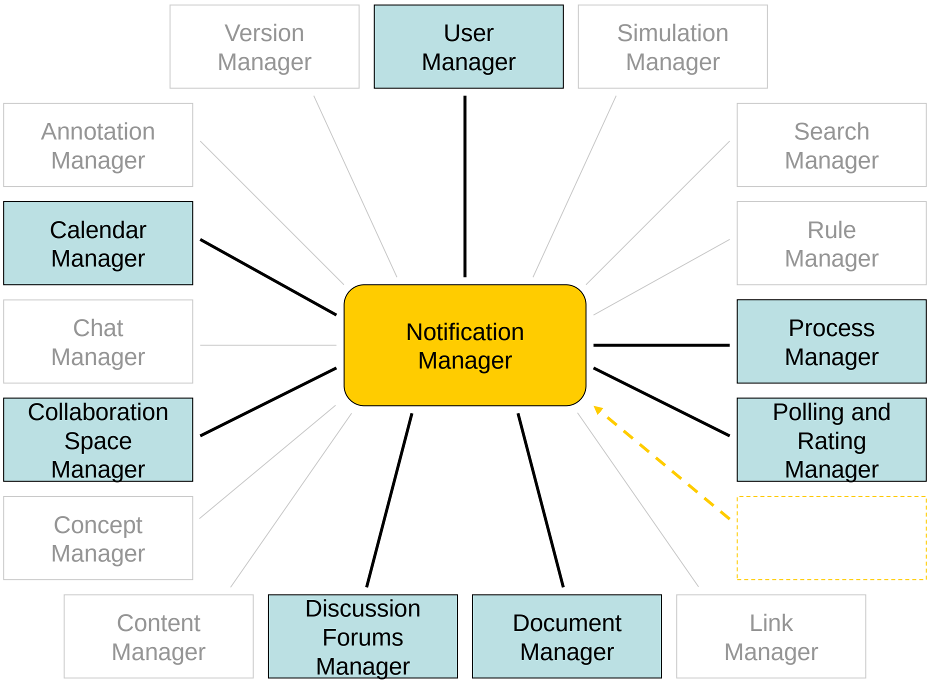


Figure 44 Context of Notification Manager

A context of the Notification Manager is depicted in Figure 44. This manager is expected to communicate with the following managers:

- Calendar manager – to support reminder functionality for calendar events and notification of their creation/modification
- Collaboration space manager – to support notification functionality for collaboration space within current project (process)
- Document manager – to support notification functionality for document management utilities (creation/modification of documents/resources)
- Discussion forums manager – to support notification functionality for discussion forums (creation/modification of discussion elements)
- Polling and Rating manager – to support polls and ratings with notification functionality
- Process manager – to retrieve process specific information playing the role of a context and support notification events of process changes within the system
- User manager – to obtain information on access rights and roles played by users

7.10.3. Supported use cases

The component provides services to other components and users for usage of notification functionality through different channels. It also supports creation of such messages (events) through user interface (or its parts) and automatically using API for other parts of the platform. In provided use case diagram Manager actor represents any manager and User actor any user-related actor that uses notification (publishing) service of Notification manager.

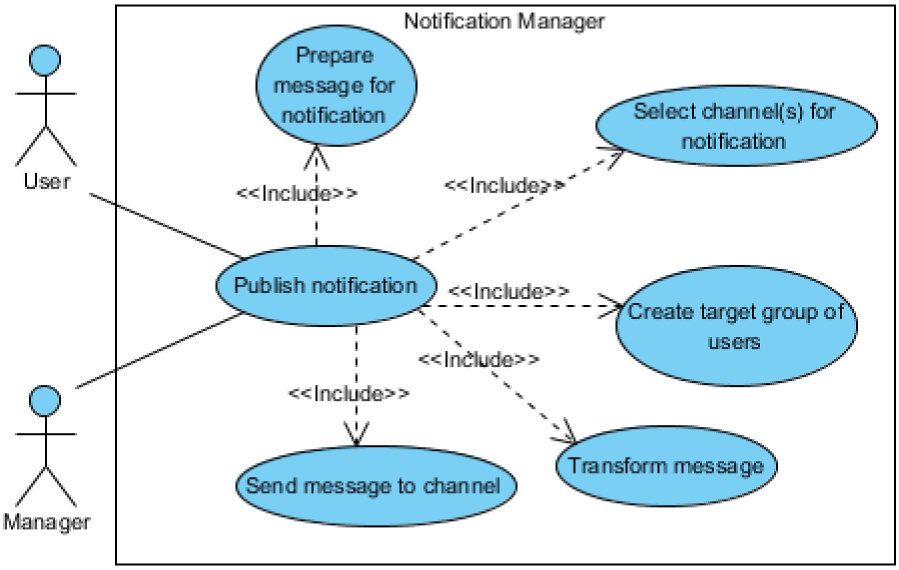


Figure 45 Use cases supported by Notification Manager

The component represents a service for notification of users using selected channels like RSS, e-mail, news, newsletter, etc. Main user-based functionality is in user interface subcodes, which can be used for preparing such messages (events) within collaboration space interfaces and select specific channels. Also it is possible (for some channels) to specify target group of users. Access rights are checked in order to differentiate usage of the notification service according to particular roles.

One part of the notification process is transformation of written notification text (from user interface or component) into suitable form for selected channels.

Another option (but in core implementation quite similar) is to provide notification API to other components. In general, also user interface part of the collaboration space for direct user notification can be seen as a 'subcomponent' of Notification manager, which only (after the moment of user input) uses API of its 'parent' component. Also in this API component can setup text for notification, select channels and target a group of users (if needed).

7.10.4. Functionality description

The component serves as a service for other components or some specific subcodes of user interfaces within the system in order to notify with new changes, actions, documents, etc. Functionality (both for users and components) includes:

- preparation of message for notification - text version specified by user or object from the data model (from which notification is specified)
- selection of channels for notification and specification of a target group of users
- transformation (automatically) of message or object into all channel-specific formats with addition of necessary context information and/or any links to resources (if applicable)
- sending (publishing) notification through selected channels to target group of users

According to current requirements it is expected to provide different channels like:

- RSS feeds
- E-mail notification
- News
- Newsletter
- Hints
- Help and assistance

All of the notification can be context-specific and its creation should be also controlled by the access rights from the user management component.

7.10.5. Component API

Function	Description
Publish notification	*To send (publish) notification message through selected channels to a specific target group of users (components).* **Input**: notification message – a message for notification (can be based on some specific data object) settings – selected channels with identified target group of users **Output**: result type – to identify whether the sending was successful or failed

Table 58 Notification Manager API

7.11. POLLING AND RATING MANAGER

7.11.1. Relevant user requirements

T-7	Opinion polling tool – open forms
T-8	Opinion polling tool – participation of users in polls – one vote per person
T-9	Opinion polling tool – participation of users in polls – possibility to modify the answers provided (versioning)
T-10	Opinion polling tool – different types of questions & answers
T-11	Opinion polling tool – presentation of the results

T-14	Discussion forums – rating of contributions and contributors (analysis of discussions based on a relevance feedback)
T-25	Commenting functionality
T-42	Tags
T-C2	News – rating/polling functionality
I-10	Opinion polling about the current version of scenario generation resources

7.11.2. Context of the component

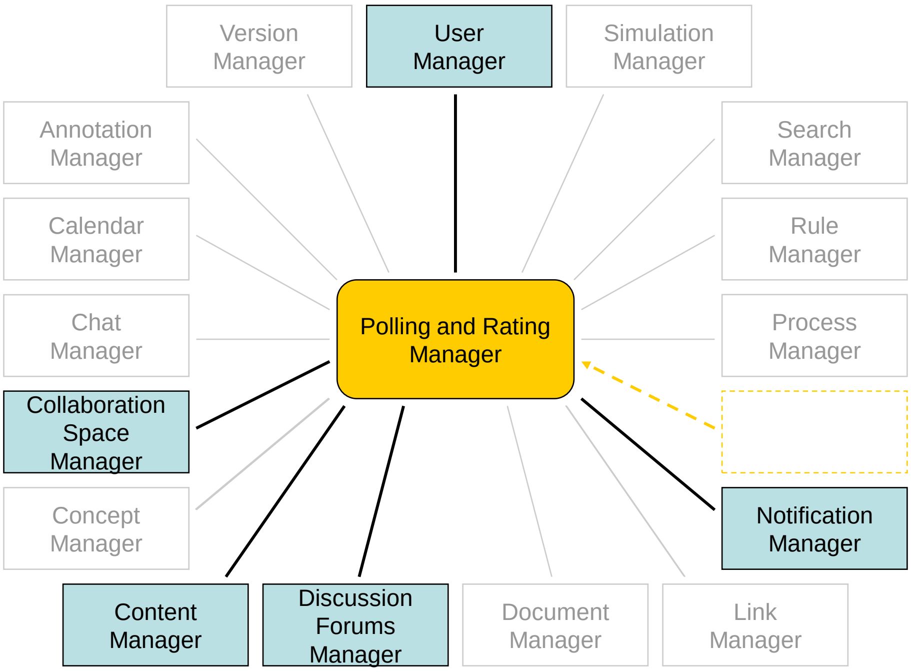


Figure 46 Context of Polling and Rating Manager

A context of the Polling and Rating Manager is depicted in Figure 46. This manager is expected to communicate with the following managers:

- Collaboration space manager – to provide polls functionality as a part of the collaboration space within the current project (process)
- Content manager – to store and retrieve data related to polls and ratings (feedbacks) to/from the content repository
- Discussion forums manager – to provide rating functionality in discussion forums (relevance feedback) and analysis based on such rating (ranking of authors)
- Notification manager – to announce polls and their results
- User manager – to obtain information on access rights and roles played by users

7.11.3. Supported use cases

The component provides services to users for creation of polls in order to retrieve users' feedback for some questions. Another type of functionality is rating of contributions within discussion forums, which is provided using API. In the presented use case diagram, User actor represents any user-related actor and Poll creator is generally any user which is authorized for creation and management of polls.

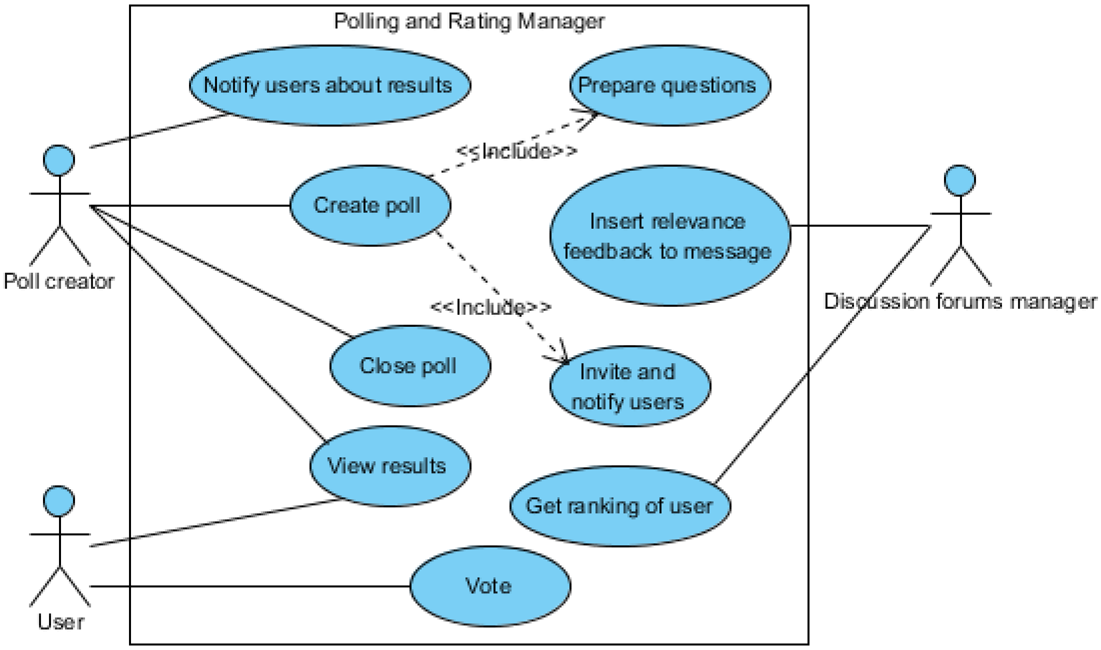


Figure 47 Use cases supported by Polling and Rating Manager

The component represents user interface for opinion polls. Users are able to create a poll by preparing question(s), inviting and notifying the users. A poll is finished according to specific settings – manually by poll creator, automatically when every invited user voted or when some amount of time for opening of poll is finished. Of course, invited users are able to vote (including setting of poll where users have more attempts for vote, if they change their mind).

Users are able to see results of the polls. Poll creator is also able to notify users about the results with its additional comments attached to them. Otherwise, system will use notification automatically with predefined presentation of results to involved users.

This component also provides service to another component – discussion forums. All users are able to provide relevance feedback to discussion forum messages (simple relevance scale, e.g. from very bad – through neutral – to very good). These feedbacks are analysed using discussion forum analysis algorithms and can be used to rank users according to feedback on their contributions. Therefore users can be more respectful if their rankings are high.

7.11.4. Functionality description

The component serves as a user interface for polls (and partially for user interface subcodes of rating), and also as a service for rating functionality (relevance feedback), especially for discussion forum messages. Therefore functionality includes:

- creation of poll – creator specifies question(s), settings for management of poll (how it is finished, if users are able to change vote while poll is active, etc.), invites users and notifies them about poll (automatic support)
- close poll – it can be closed in different ways, e.g. manually by poll creator, automatically when every invited user voted or after some time deadline
- vote – user is able to vote, if it is possible then he/she can change vote during active poll
- view results – results are only generated automatically and system can notify the user
- notify users about results – creator can add some comments and interpretation to automatically created results and then notify the users manually
- rating functionality – manager provides rating service to discussion forums in order to make relevance feedback about contributions of users
- ranking of users – discussion analysis allows the component to provide rank of the users (absolute or relative) according to feedbacks on their own messages (can be extended using subsets of users and topics)

It is expected that polls have more modes of work and functionality (setup by settings), e.g. different types of questions/answers, multiple choices, edit vote function (if needed). For storage and retrieving of manager-specific data content repository is used. Context is (intentionally) shared using collaboration space and its personalisation and/or customisation to current process. All actions are also controlled by the access rights from the user management component.

7.11.5. Component API

Function	Description
Insert relevance feedback	*To rate the message in discussions with relevance feedback.* **Input**: message – an identification of a message feedback – value of relevance chosen by user **Output**: result type – to identify whether the rating action was successful or failed
Get ranking of user	*To obtain the rank of selected user according to relevance feedback on his/her messages.* **Input**: user – an identification of a user **Output**: ranking – user’s authority according to discussion analysis in form of absolute or relative ranking number

Table 59 Polling and Rating Manager API

7.12. PROCESS MANAGER

7.12.1. Relevant user requirements

I-1 ICT toolbox functionality provided through one portal-based interface

I-2	Transformation table – connection of context-specific information within the Scenario Generation and Policy Modelling process in ICT toolbox
I-3	Starting the scenario generation process - initial scenario
I-11	Closing the scenario generation process / versioning
I-4	Creation of stakeholder groups for the scenario generation process
I-7	Integration of components within the e-participation tools for scenario generation – workspace
I-10	Opinion polling about the current version of scenario generation resources
I-12	Support for direct export/import of information between scenario generation process and policy modelling
I-13	Control of Scenario Generation process phases
I-15	Support for the policy modelling tool to create a new scenario generation iteration
I-23	Creation of stakeholders groups for policy modelling process
I-19	Log of activities within scenario generation
I-20	Log of activities within policy modelling / simulation
I-32	Workflow support
I-F-I6	Personalise overview

New requirements:

SOTA-1	Workflow engine
UC-1	Rights management
UC-2	Invitation – send and receive
UC-3	Send request for invitation
UC-4	Initiate project

7.12.2. Context of the component

A context of the Process Manager is depicted in Figure 48. This manager is expected to communicate with the following managers:

- Annotation manager – to provide process-specific information to particular manager and obtain current process-specific changes
- Collaboration space manager – to provide process-specific information to particular manager and obtain current process-specific changes; to reflect current process status in collaboration space settings/features
- Concept manager – to provide process-specific information to particular manager
- Content manager – to store and retrieve information related to process/workflow management (current process status, process definition/flow, etc.)
- Link manager – to provide process-specific information to particular manager
- Notification manager – to publish/notify users about process changes and status (and related artefacts) through different channels (email, news, newsletter, RSS, etc.)
- Rule manager – to provide process-specific information to particular manager
- Search manager – to provide process-specific information to particular manager; to support search in process-specific information (if needed)
- Simulation manager – to provide process-specific information to particular manager

- User manager – to obtain information on access rights and roles played by users
- Version manager – to support versioning in storage and retrieving information related to process management

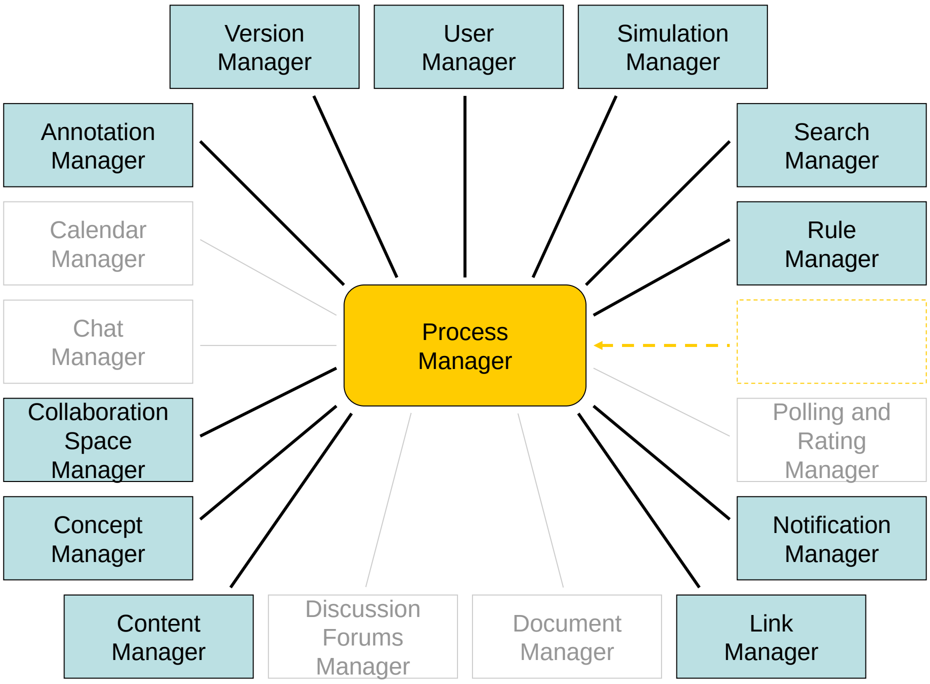


Figure 48 Context of Process Manager

7.12.3. Supported use cases

The component mainly provides services to other components, it represents a part of the system responsible for process definition and status, playing the role of a context within the platform. In provided use case diagram Manager actor represents any manager that uses context service of Process manager. User (Initiator) is responsible for creation of a new project (process), User (Workflow) is any user/actor which is able to manage workflow steps. User-related steps will have their own user interface within the application.

The main objective is to provide context information about the current process and its status. According to this information most of the other components customise their work (especially collaboration space). Therefore context information should be provided as a service to other components.

The process initiation (creation of a project and respective shared space) is also supported within manager's use cases. Process management then controls running process(es) and makes all context information stored in correct form. Executing of processes is expected to be simple and only includes activities available within the platform (without any external services or complex subprocesses). Process-specific changes can be obtained also from other components.

One of the process definition elements is to provide context-specific access rights, but this will be done using User manager with context information from Process manager. Of course, access rights for management of process itself should be also defined there.

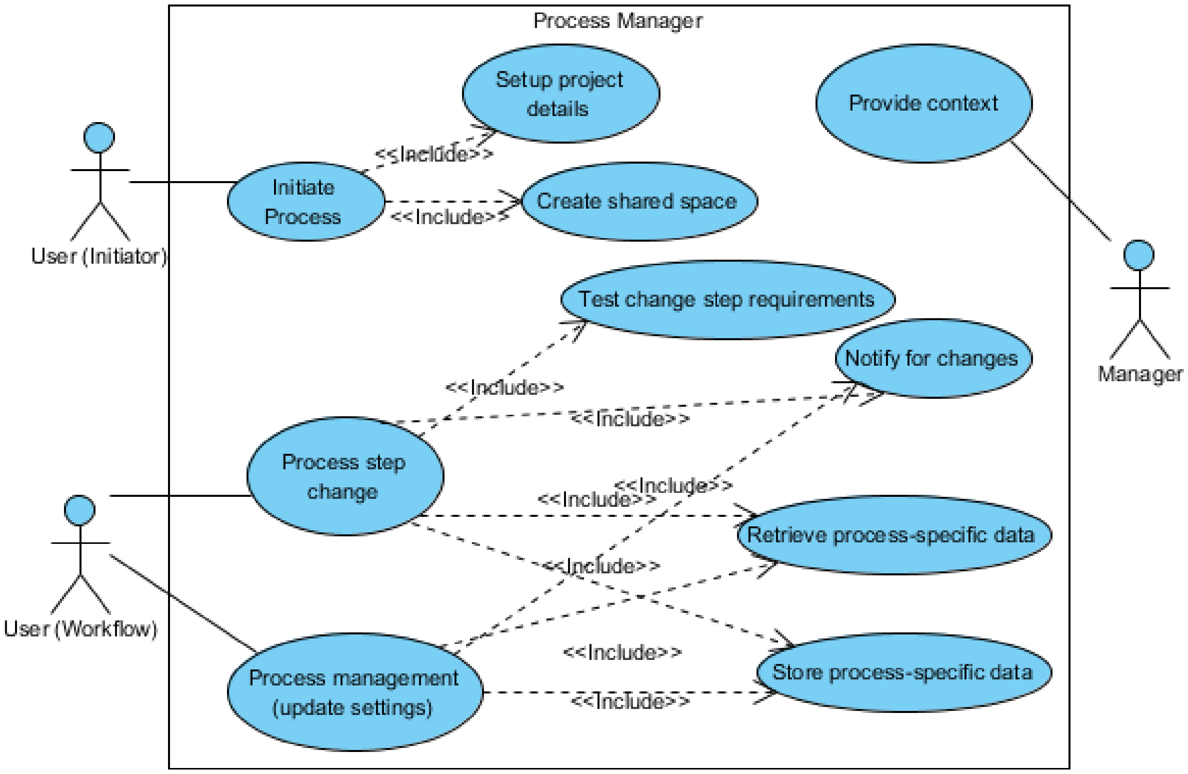


Figure 49 Use cases supported by Process Manager

7.12.4. Functionality description

The component serves as a context owner and provider. It is also possible to manage process using this manager, but it is expected to have static composition of workflow activities, prepared only for identified basic process definition (policy modelling process). This will be done off-line in design phase of the platform directly and manually for supported policy modelling process. It includes:

- definition of correct initiation of the process (project)
- definition of basic states of the workflow (state entities)
- definition of requirements for changing between states (input/output of tasks or activities)
- description of available tools and functionality (and details regarding usage of collaboration space and repositories) within particular steps

The current process status is then available as context information (process, current state, general process/state settings, specific process/state settings) and is provided to other components for their own customisation and usage. Some simple mechanism should be supported with basic workflow definition support, where human tasks and readability are well matched (e.g. JBPM from JBoss).

Information related to process management is stored and retrieved using content repository and can be versioned according to process state changes. Content repository objects of process-specific data are also available for search, but in direct connection to repository utilities (Content manager), therefore service provided by the process component is not needed.

7.12.5. Component API

Function	Description
Initiate process	*To initiate new process instance in order to start new process execution.* **Input**: process definition – a definition of process for initiation with necessary general and specific details **Output**: process instance – an instance of the process with formal description of the current context
Provide context	*To provide context (current process) information to other components.* **Input**: process instance identification – an identification of a current process for retrieving correct instance details **Output**: context – an object with current process details

Table 60 Process Manager API

7.13. RULE MANAGER

7.13.1. Relevant user requirements

FR01_PM PM (Transformation process) - Define initial policy modelling aspects

FR02_PM PM (Transformation process) - Stakeholder extraction

FR03_PM PM (Transformation process) - Environment generation

FR04_PM PM (Transformation process) - Goal definition

FR05_PM PM (Transformation process) - Rule generation

FR06_PM PM (Transformation process) - Assumption definition

FR07_PM PM (Modelling process) - Agent type creation

FR08_PM PM (Modelling process) - Agents at different aggregation levels

FR09_PM PM (Modelling process) - Exogenous factors

FR10_PM PM (Modelling process) - Environment definition – general

NFR02_PMPM (Transformation process) - Language transition

NFR04_PMPM (Modelling process) – End states

NFR04_PMPM (Modelling process) – Initial model definition (Beginner's mode)

NFR05_PMPM (Modelling process) – Iterations (Expert's mode)

NFR06_PMPM (Modelling process) – Model description

NFR07_PMPM (Modelling process) – General model description

I-12 Support for direct export/import of information between scenario generation process and policy modelling

I-14	Maintaining of scenarios and rules within the ICT toolbox
I-25	Integration of policy modelling tool and simulation / analysis tools – data exchange / annotation
I-30	Translation of agent rules from a tool neutral syntax into simulation back-end language
I-40	Transition table browser

New requirements:

SOTA-6	Information structuring
UC-9	Network visualisation
UC-10	Development of social network

7.13.2. Context of the component

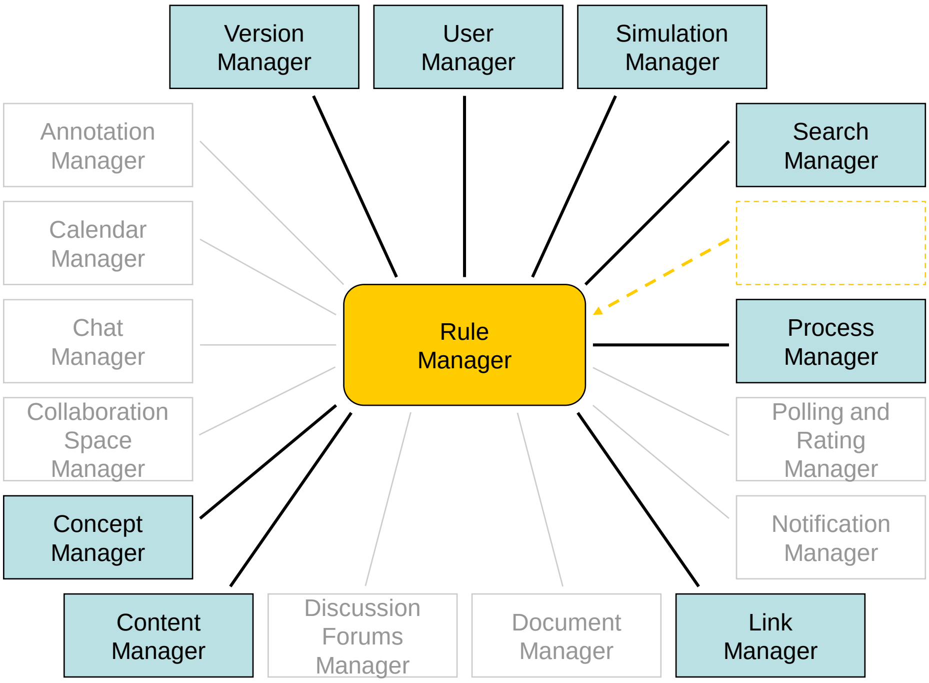


Figure 50 Context of Rule Manager

A context of the Rule Manager is depicted in Figure 50. This manager is expected to communicate with the following managers:

- Concept manager – to retrieve and store conceptual knowledge structures (e.g. hierarchies, tuples, dependencies and networks, etc.)
- Content manager – to store and retrieve modelling elements (agents, fact templates, rules, etc.)
- Link manager – to store information on relations between objects (knowledge structures and/or modelling elements) and to search for objects related to a given object.
- Process manager – to retrieve process specific information playing the role of a context
- Search manager – to search for an object (knowledge structure or modelling element)
- Simulation manager – to export formal knowledge bases

- User manager – to obtain information on access rights and roles played by users
- Version manager – to store and retrieve information on different versions of objects

In addition to this, it is possible to utilise also other managers: Document manager for publishing knowledge bases; Polling and Rating manager as well as Chat and Discussion forums communication managers for creating a communication channel linked to (a set of) objects managed by the Rule manager.

7.13.3. Supported use cases

The component mainly provides services to users, it represents a part of the system's front-end user communicates with. Its aim is not to support other components (or only in a small extent). In the presented use case diagram, User actor represents any user-related actor that uses functionality of Rule manager.

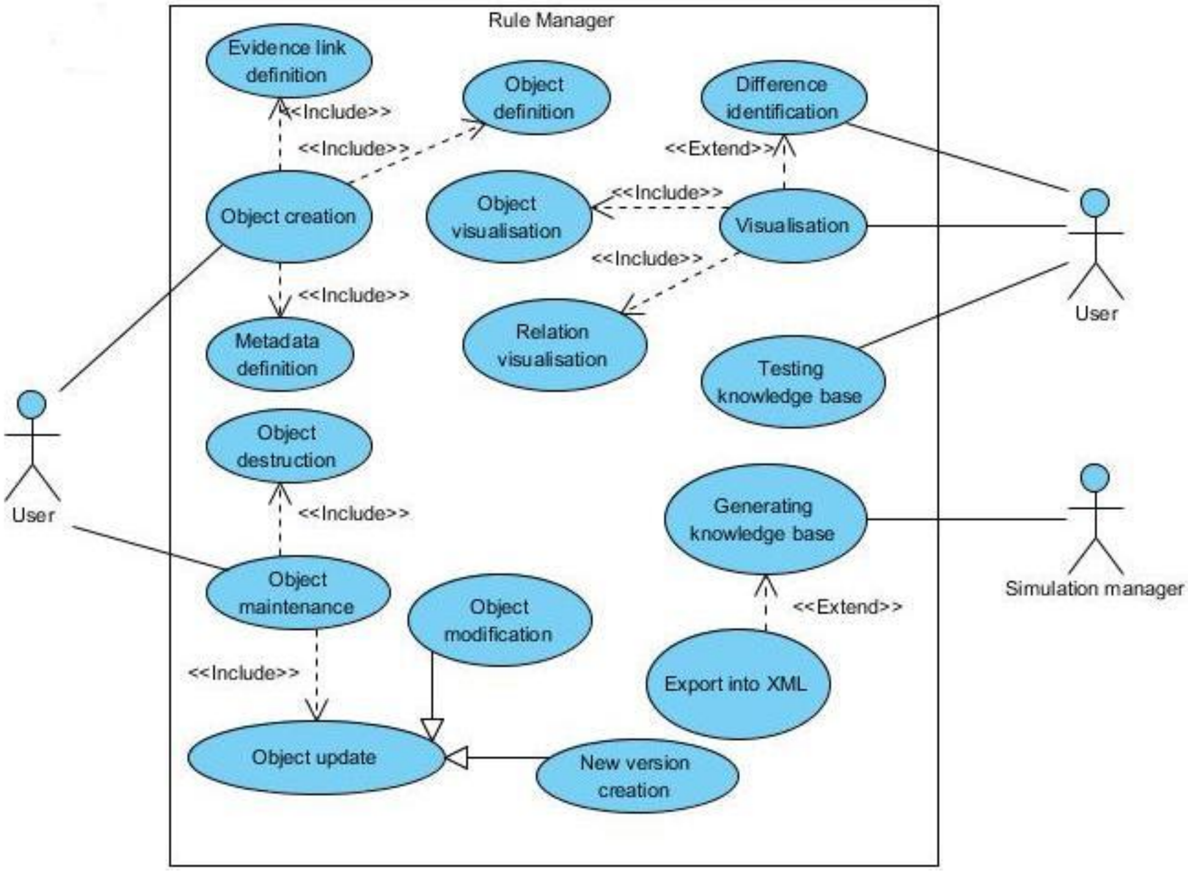


Figure 51 Use cases supported by Rule Manager

The component basically represents a specialised 'editor' dedicated to dealing with special objects (knowledge structures, modelling elements and knowledge bases). The presented use cases indicate processing and manipulations with mentioned objects. This term should be subclassed (not shown due to space limitation) – to represent the processed object types.

It is possible to create objects (objects themselves, their metadata and links to represent evidence for the objects). In a similar way, objects can be updated or destroyed.

All created objects can be visualised. It is possible to visualise not only objects themselves but follow relations between objects and move back and forth between objects and their evidences. Moreover, different versions of objects can be compared and their differences can be visualised as well.

Semiformal object representations can be transformed into a formal representation of knowledge bases. This formal representation enables to perform different kinds of tests to ensure required characteristics of generated formal knowledge bases.

7.13.4. Functionality description

The component serves as a part of a bridge connecting evidence-based scenarios with formal models. Its aim is threefold:

- to further elaborate and organise knowledge structures identified during scenario analysis
- to transform these knowledge structures into semiformal representation of modelling elements
- to generate formal knowledge bases from the semi-formal elements

In order to define main information and knowledge chunks, which form a base for subsequent modelling activities, conceptual knowledge structures should be restructured and transformed. It is expected that the following semiformal structures will be identified:

- hierarchies of objects (e.g. actor hierarchies)
- object-attribute-value tuples (e.g. actor skills)
- dependencies and networks (e.g. actor social network)
- priorities and orderings (e.g. actors)
- annotations (e.g. endorsement characteristics)

To support the creation and maintenance of these knowledge structures, the component enables basic functionality to create them, modify as well as destroy them. Since the structures can be present in different versions, it is possible to compare them, indicate differences and provide an analysis if applicable (e.g. analysing identified social networks).

Knowledge structures should be turned into basic modelling elements. The production of the defining elements of formal models (agents, facts, fact templates, RHS/LHS clauses, conditions, rules) is supported on a level of creating, updating and deleting. In addition to modelling elements based on acquired knowledge structures, it is possible to insert elements not based on evidence (magic elements for which an expert/modeller serves as a reference since the elements are based on his/her intrinsic knowledge).

The defined modelling elements, which are represented semiformally (e.g. using a pseudocode), can be transformed into a formal knowledge base (or bases – one for each agent) and exported in an XML format. In order to support the production of high quality bases, different kinds of analysis are expected to be performed on the knowledge base (e.g. redundancy check, consistency and reachability analysis).

All elements, the component manipulates with (knowledge structures, modelling elements, knowledge bases) are expected to be interlinked in order to preserve the 'cause – result' relationship. If an object (knowledge structure, modelling element) has an evidence/endorsement, then it will be linked to it. These links enable visualisation of the dependence among different objects, enabling going back and forth in this network (e.g. going from a formal rule in a knowledge base to semiformal modelling elements and from a modelling element to supporting knowledge structures).

In order to support different views and/or different evolution stages, all elements can be versioned (different versions of an object can exist) and versions of the same object can be compared to indicate differences.

7.13.5. Component API

Function	Description
Knowledge base generation	*To initiate transformation of semiformally represented modelling elements into a formal knowledge base, respecting syntactical requirements of target simulation rule engine.* **Input**: target engine – an identification of a simulation engine which is expected to process the formal knowledge base tag – an identification of a scope of generated knowledge base (only those modelling elements are considered which comprise the tag in their metadata) **Output**: knowledge base – a formally represented knowledge base expressed in the required syntax

Table 61 Rule Manager API

7.14. SEARCH MANAGER

7.14.1. Relevant user requirements

I-1	ICT toolbox functionality provided through one portal-based interface
I-6	Integration of components within the e-participation tools for scenario generation – search
I-5	Integration of components within the e-participation tools for scenario generation – data exchange / annotation
I-7	Integration of components within the e-participation tools for scenario generation – workspace
I-19	Log of activities within scenario generation
I-20	Log of activities within policy modelling / simulation
I-F-I6	Personalise overview

7.14.2. Context of the component

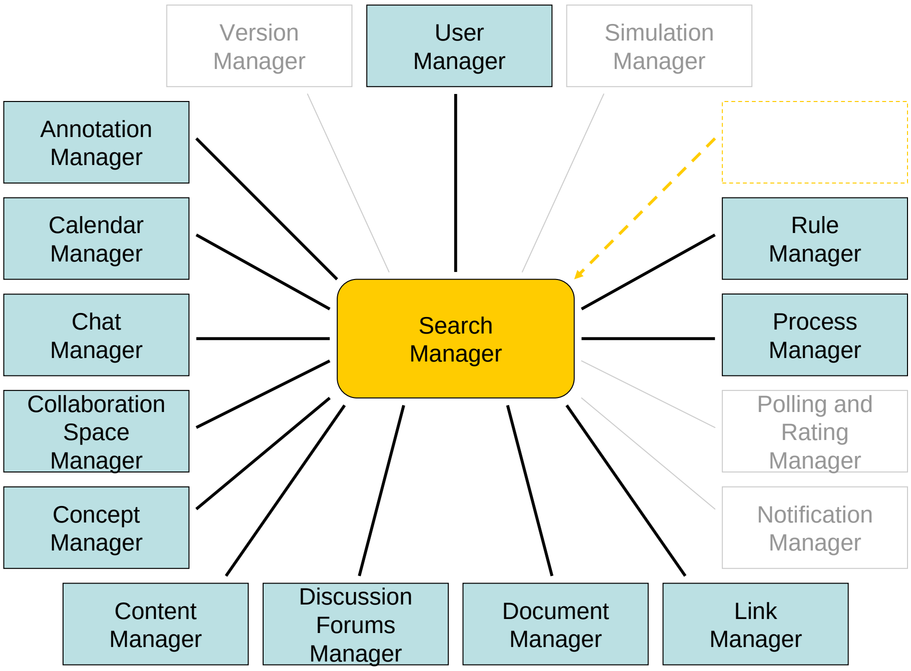


Figure 52 Context of Search Manager

A context of the Search Manager is depicted in Figure 52. This manager is expected to communicate with the following managers:

- Annotation Manager – to search in annotation elements/objects from data available within the scenario analysis tool
- Calendar manager – to search in calendar-specific data
- Content manager – to search directly in content repository (if needed), especially in specific metadata; can be used also to store and retrieve search-specific data (stored queries, search settings, etc.)
- Chat manager – to search in chat messages from the previous chat sessions; Document manager used to retrieve old chats
- Concept manager – to search in structured model for evidence-based conceptual descriptions of problem data (known as CCD)
- Collaboration space manager – to allow and provide federated search in specific user interface part of the collaboration space
- Document manager – to search in documents within the system
- Discussion forums manager – to search in discussion forums
- Link manager – to search in specific metadata about linking objects within the data in system (where links are specific conceptual objects for support of evidence-based modelling that connect CCD elements, data sources, simulation models, etc.)
- Process manager – to retrieve process specific information playing the role of a context
- Rule manager – to search in knowledge structure or modelling element (within rule bases, models, etc.)
- User manager – to obtain information on access rights and roles played by users

7.14.3. Supported use cases

The component provides services for search in different data types / applications with possibility to forward them and combine in specific user interface within collaboration space or to show them individually per tools. In provided use case diagram Manager actor represents any manager that uses search service of Search manager. User actor represents any user-related actor that uses functionality of Search manager.

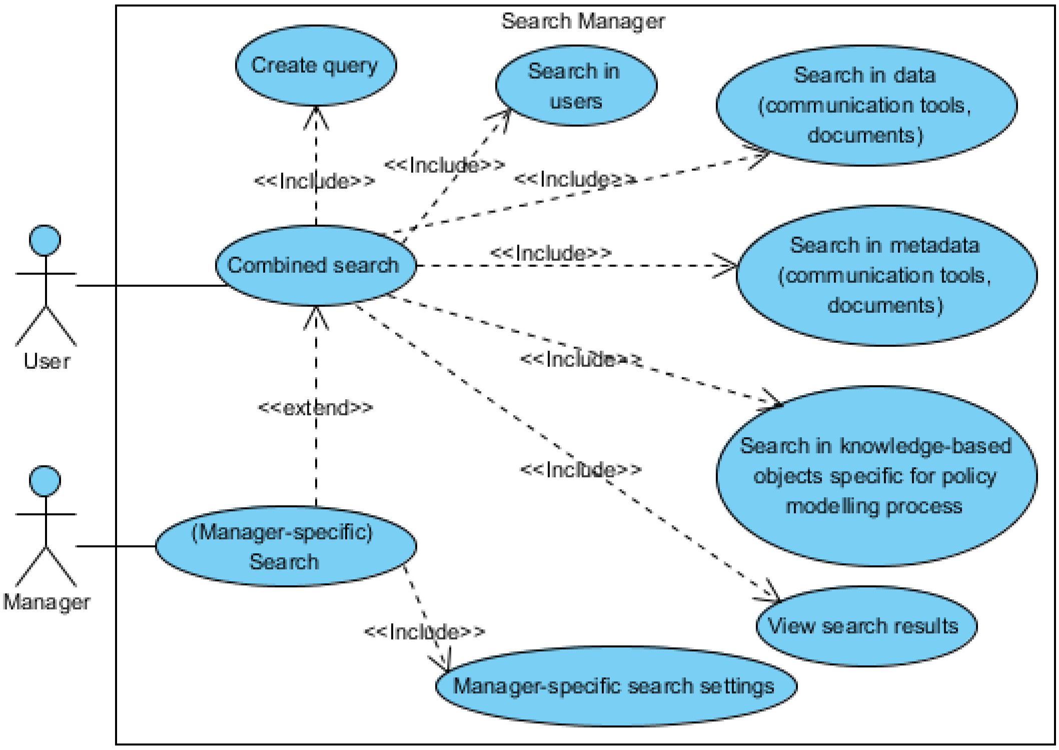

Search Manager
Create query
Search in users
Search in data (communication tools, documents)
Search in metadata (communication tools, documents)
Search in knowledge-based objects specific for policy modelling process
View search results
Manager-specific search settings
Combined search
(Manager-specific) Search
User
Manager
<<Include>>
<<Include>>
<<Include>>
<<Include>>
<<Include>>
<<Include>>
<<Include>>
<<extend>>

Figure 53 Use cases supported by Search Manager

The Search Manager supports all searching aspects in OCOPOMO. It is expected that search manager helps in providing single entry point for all OCOPOMO's search needs. Different types of data are searched in different manner:

- Search in users (collaboration space manager)
- Search in data within communication tools and document library
- Search in metadata related to communication tools and document library
- Search in knowledge-based objects specific for policy modelling process (annotations, concepts, links, etc.)

7.14.4. Functionality description

The component serves as a general search module for supporting other components. Main aim of the component is to provide search service in more transparent and interoperable way, while particular searches can be run in different sessions, databases and repositories.

Search in users will be implemented within the collaboration space manager specific data (can be stored in content repository).

Search in data within communication tools and document library can be available using application (tools) specific data storage (if they have their own mechanism). Content repository (managed by Content manager) will be used (where applicable).

Search in metadata and specific knowledge modelling elements (concepts, links, annotations) will be available by the implementation or reuse of content repository services for search. User interface for combined search is a part of the collaboration space (as other tools).

The following features are expected: 1.) The caller has to be able to define which type of search should be carried out; 2.) It has to be possible to determine the best search type automatically by evaluating the amount of available input data and other information concerning the search request.

7.14.5. Component API

Function	Description
Search	*Search OCOPOMO's sources according to a given request* **Input**: query – either a keyword-based simple query or a query prepared for semantic matching (containing requirements, capabilities, ...) filters – constraints for filtering the search results in addition to automatic filtering **Output**: hits – matched data model elements

Table 62 Search Manager API

7.15. SIMULATION MANAGER

7.15.1. Relevant user requirements

T-16	Agent-based simulation tool
T-17	PM (Analysis) - Export of simulation-related data
T-18	Import of the previously exported simulation data
T-19	Previewing of a simulation
T-20	Preview simulation mode – level of details and/or time scale
T-21	Preview simulation mode – searching for a specified event
T-22	Preview simulation mode – focusing on a part of the used model
T-23	PM (Analysis) - Qualitative representation of the simulation results
T-32	PM (Gaming) – Role-playing games (single user)
T-33	PM (Gaming) – User interface for human player
FR11_PM	PM (Simulation setup) - Setup world facts
FR12_PM	PM (Simulation setup) - Setup initial agent facts

FR13_PM	PM (Simulation setup) - Initial state definition
FR14_PM	PM (Simulation termination) - End state
FR15_PM	PM (Simulation termination) - Irregular termination events
FR16_PM	PM (Simulation termination) - Regular termination events
FR17_PM	PM (Simulation termination) - Adjustable parameters
FR18_PM	PM (Simulation termination) - State validation
FR19_PM	PM (Simulation termination) - Simulation start
FR20_PM	PM (Simulation termination) - Simulation interrupt
FR21_PM	PM (Simulation termination) - Simulation abort
FR22_PM	PM (Experimentation) - User engagement in simulation
FR23_PM	PM (Experimentation) - User Interaction
FR24_PM	PM (Experimentation) - Gaming simulation interface
FR25_PM	PM (Experimentation) - Change simulation parameters
FR26_PM	PM (Experimentation) - Automated experimentation
FR27_PM	PM (Gaming) - Feedback on simulation
TP-1	PM (Analysis) – Within-timestep dependency graph visualisation
TP-2	PM (Analysis) – Experiment and rule development browser
TP-3	PM (Analysis) - Narrative output
TP-5	PM (Analysis) - Visualisations of non-numerical outcomes/events
NFR08_PM	PM (Simulation) – Event handling
NFR09_PM	PM (Simulation) – Exception handling
NFR10_PM	PM (Simulation) – Simulation visualisation
NFR11_PM	PM (Simulation) – Parameter presentation
NFR12_PM	PM (Simulation) – Parameter locking
NFR13_PM	PM (Simulation) – State handling for inspection
NFR14_PM	PM (Simulation) – Simulation execution
I-18	Comparison of simulations
I-22	Defining scenario for policy modelling
I-27	Simulation preview tool available from different physical locations – remote access
I-28	Action–based and rule-based role playing of stakeholders in simulation
I-20	Log of activities within policy modelling / simulation
I-29	Human actions analysis
I-34	Simulation back-end integrated with the ICT toolbox
I-39	Full dependency graph including dependency of rules on lagged clauses

New requirements:

SOTA-8	Non-RETE rule engine

7.15.2. Context of the component

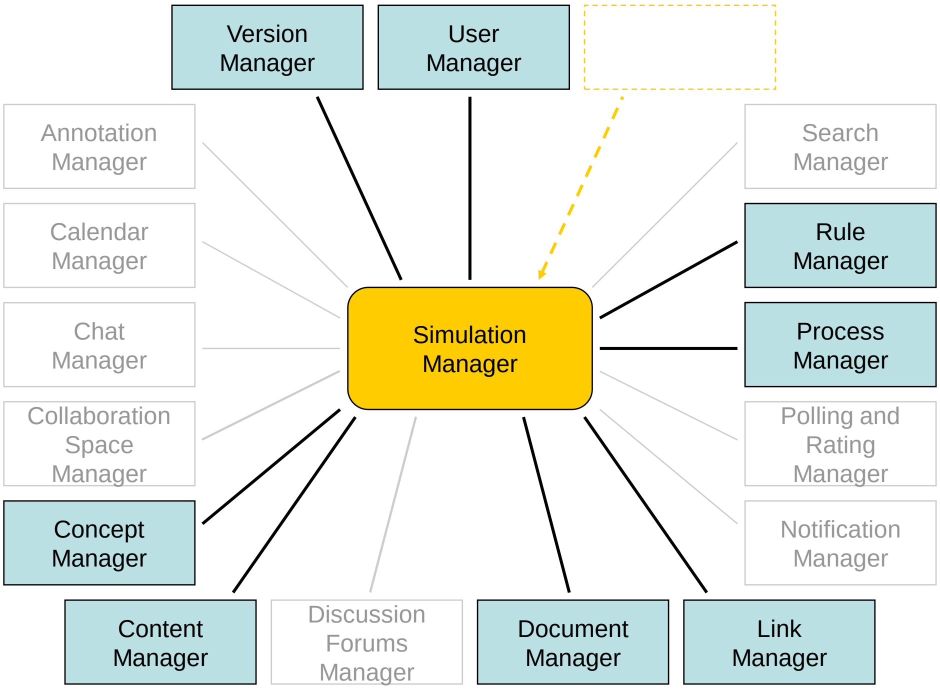


Figure 54 Context of Simulation Manager

A context of the Simulation Manager is depicted in Figure 54. The manager is expected to communicate with the following managers:

- Concept manager – to store modelling elements (e.g. new or modified rules)
- Content manager – to store and retrieve simulation related data (e.g. simulation context, current simulation state)
- Document manager – to publish model-based scenarios
- Link manager – to store information on relations between objects (fact/rules and dependency graphs, dependency graphs and simulation outputs). To search for objects related to a given object.
- Process manager – to retrieve process specific information playing the role of a context
- Rule manager – to obtain formal knowledge bases
- User manager – to obtain information on access rights and roles played by users
- Version manager – to store and retrieve information on different versions of objects

In addition to this, it is possible to utilise also other managers: Polling and Rating manager as well as Discussion forums manager and Chat manager for creating a communication channel linked to simulation outputs.

7.15.3. Supported use cases

The component mainly provides services to users, it represents a part of the system's front-end user communicates with. Its aim is not to support other components. In the presented use case diagram, User actor represents any user-related actor that uses functionality of Simulation manager. <<System>> is any OCOPOMO system/process component that uses import/export functionality of Simulation manager.

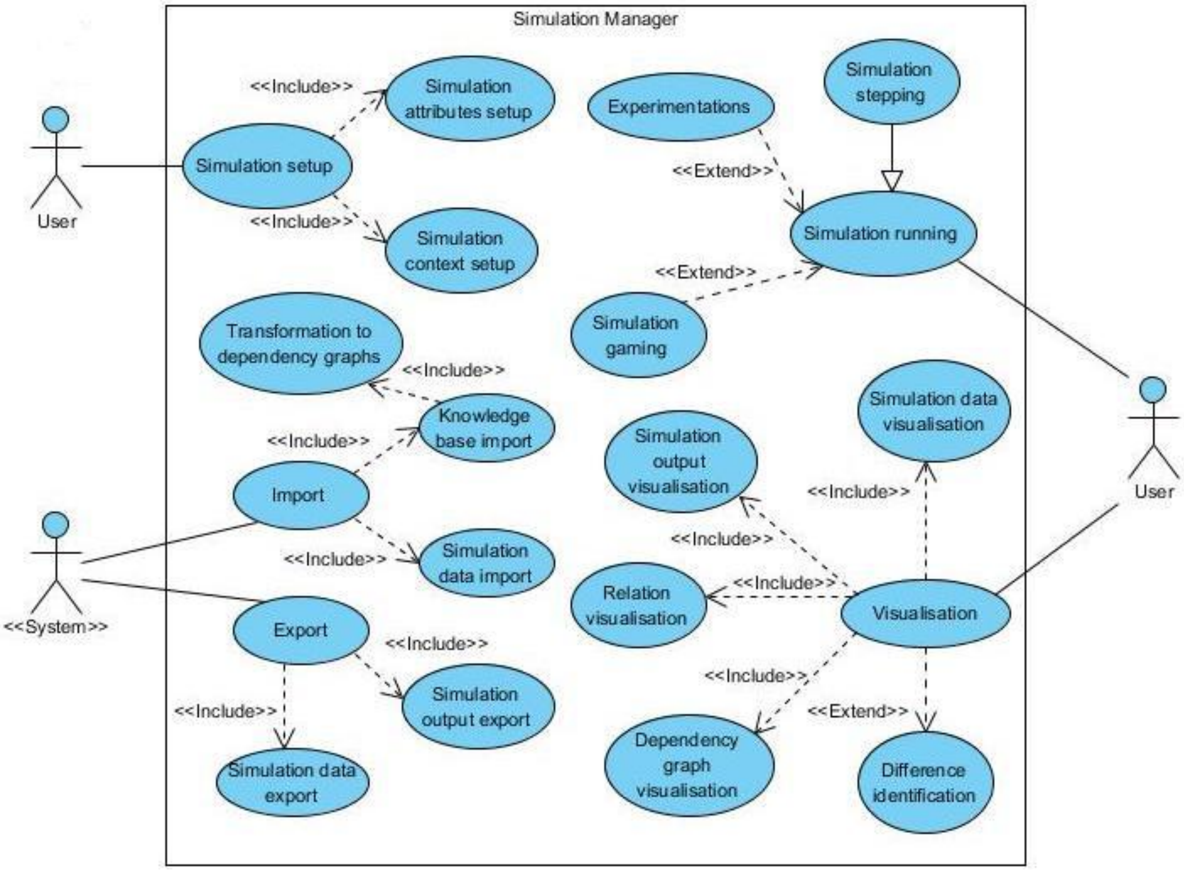


Figure 55 Use cases supported by Simulation Manager

The core of the manager is a simulation engine able to run formal models utilising agent-based and rule-based knowledge bases. It enables to run plain simulations (based on actual simulation setups) as well as provides a rich set of opportunities to experiment with models including simulation gaming when users play the role of agents.

This simulation engine is supported on both input and output sides. On the input hand, a knowledge base is imported and transformed into data/rule dependency graphs on which the simulation engine is able to operate. On the output hand, simulation output can be processed (visualised or exported for further processing).

To support understanding about what is going within simulations as well as what relations can be found between data chunks the manager manipulates with, rich visualisation functionality is offered. It is possible to visualise artefacts as dependency graphs and simulation outputs as well as simulation process and simulation status. In addition, it is possible to make comparisons between objects, identify differences and visualise these differences. To understand mutual dependences, visualisation of relations between objects is at the disposal.

7.15.4. Functionality description

The manager employs formal knowledge bases produced by the Rule manager. The knowledge bases can be accepted in two formats – in XML as well as in the form of a programming code suitable for the given manager.

In order to utilise a knowledge base, it must be converted into the form of a data/rule dependency graph. To support users in understanding knowledge bases, it is expected that the manager provides visualisation support – a possibility to graphically visualise data/rule dependency graphs as well as to compare two dependency graphs, identify differences and visualise them.

The main functionality of the manager is the ability to run simulations. First, it is possible to setup a simulation context (e.g. to define an initial state of all agents as well as the state of simulation environment, select agents to be involved in simulation, setup world facts, stop conditions, etc.) as well as simulation attributes (e.g. time granularity, termination events, etc.). After the setup is finished, simulation can run. In addition to simple starting and running a simulation, a set of richer control possibilities will be provided to users to experiment with the models, for example:

- interrupting simulations
- stepping simulations
- adjusting parameters
- restarting / resuming simulations
- export / import of simulation data

In order to support experimentation, simulations must be visualised (e.g. visualisation of current state of agents' fact bases, current data dependency graph). To support users in testing and enhancing dependency graphs, a possibility to experiment on-the-fly with rules (e.g. enabling/disabling rules, modifying rules, adding rules) should be available as well. Another kind of experimentation is simulation gaming (action-based or rule-based) – a user takes over the role of an agent and responds on behalf of the agent according to his/her own mental model. Analysis of user's behaviour can subsequently serve as a useful input material for modellers.

Simulation runs are expected to produce two types of output:

- audit trail – basically a log of all events which occur during a simulation run (with the possibility to define which aspects are relevant enough to be logged)
- model-based scenario – a scenario produced as a text-based descriptions of actions of agents and some account of the reasons for those actions (published using Document manager API)

To support users in understanding these outputs, the manager provides e.g. functionality enabling to work with the outputs easier (e.g. different filtering levels based on the required level of details, reduction of voluminous output to the main aspects, focusing on a part of the used model, selecting a proper time scale, searching for a specific event, considering agents at different aggregation level, qualitative representation, etc.). Comparing outputs of two simulation runs and identification of differences between them is considered useful for users as well. In order to make a statistical analysis of the produced outputs, the manager enables to export simulation outputs into a form accepted by an external statistical package (currently the R package is considered).

A very useful feature enabling to understand relations between different artefacts used within the modelling process is a possibility to move back and forth between events in simulation output and data dependency graph as well as between (parts of) data dependency graph and elements of formal knowledge base. This will complement similar functionality provided by Rule manager.

7.15.5. Component API

none (the manager does not provide functionality for other managers)

7.16. USER MANAGER

7.16.1. Relevant user requirements

T-1-4	Discussion forums - Authorisation on level of the discussion forum
T-37	Authorization/authentication issues are taken into account in individual tools
I-F-I1	Password reminder
I-F-I2	Removing profile
I-F-I3	Login
I-F-I4	User registration
I-F-I5	User profile
I-F-I6	Personalise overview
I-36	All personal preferences in one place
I-NF-4	Authentication
I-NF-5	Authorization
I-NF-6	Privacy
New requirements:	
UC-1	Rights management
UC-2	Invitation – send and receive
UC-3	Send request for invitation

7.16.2. Context of the component

A context of the User Manager is depicted in Figure 56. This manager is expected to communicate with the following managers:

- Annotation manager, Calendar manager, Chat manager, Concept manager, Discussion forums manager, Document manager, Link manager, Notification manager, Polling and Rating manager, Rule manager, Search manager, Simulation manager – to provide information on access rights and roles played by users
- Process manager – to provide information on access rights and roles played by users; to retrieve process specific information playing the role of a context

- Collaboration space manager – to provide information on access rights and roles played by users; to support profile management for setting of collaboration space elements

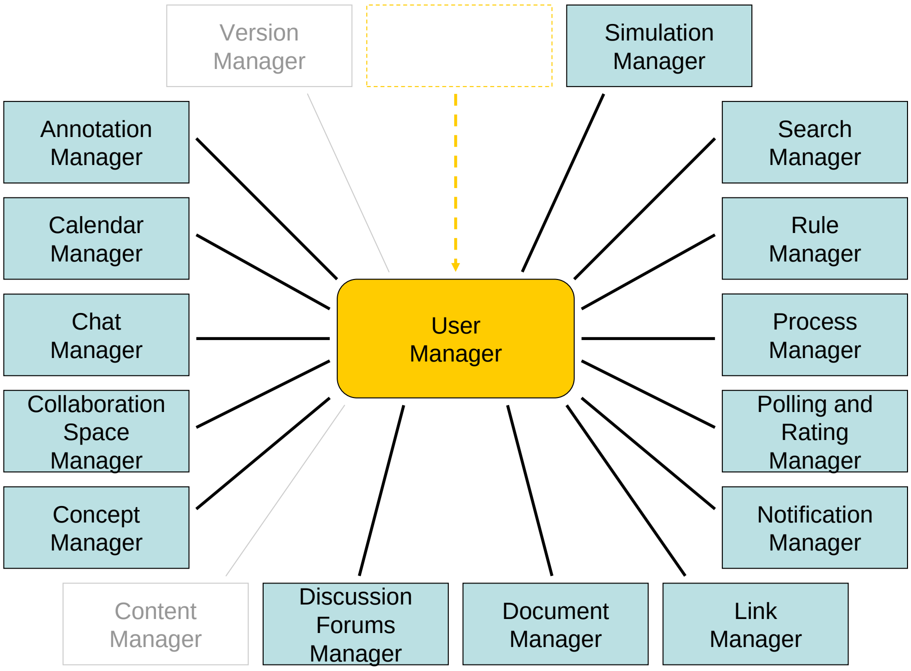


Figure 56 Context of User Manager

7.16.3. Supported use cases

The component provides service for components to obtain access rights about the usage of the functionality by some specific user and/or role. Other important issue is user registration and login, as well as user profile management (which can be reused by other components for personalisation and customization). In provided use case diagram Manager actor represents any manager that uses services of User manager for checking access rights and profile management. Administrator represents an actor with granted permission to modify user’s role(s) and access rights. User actor is generally a registered user. Unregistered user actor is a user which is from outside of the OCOPOMO site (he/she wants to be registered in order to become User). Role-based access control (RBAC) will be used for access rights management.

The component basically represents a service provider that enables to manage and retrieve access rights to (almost) all components for their actions. There is also possibility to setup some access rights dynamically using roles (where applicable, e.g. collaboration space, discussion forums, etc.), but most of the actions are strictly defined in design time (using predefined roles for process and tool specific actions/operations).

Non-registered user is also able to register (due to invitation) and get involvement within the system. Administrator is able to setup access rights (using roles) to new users.

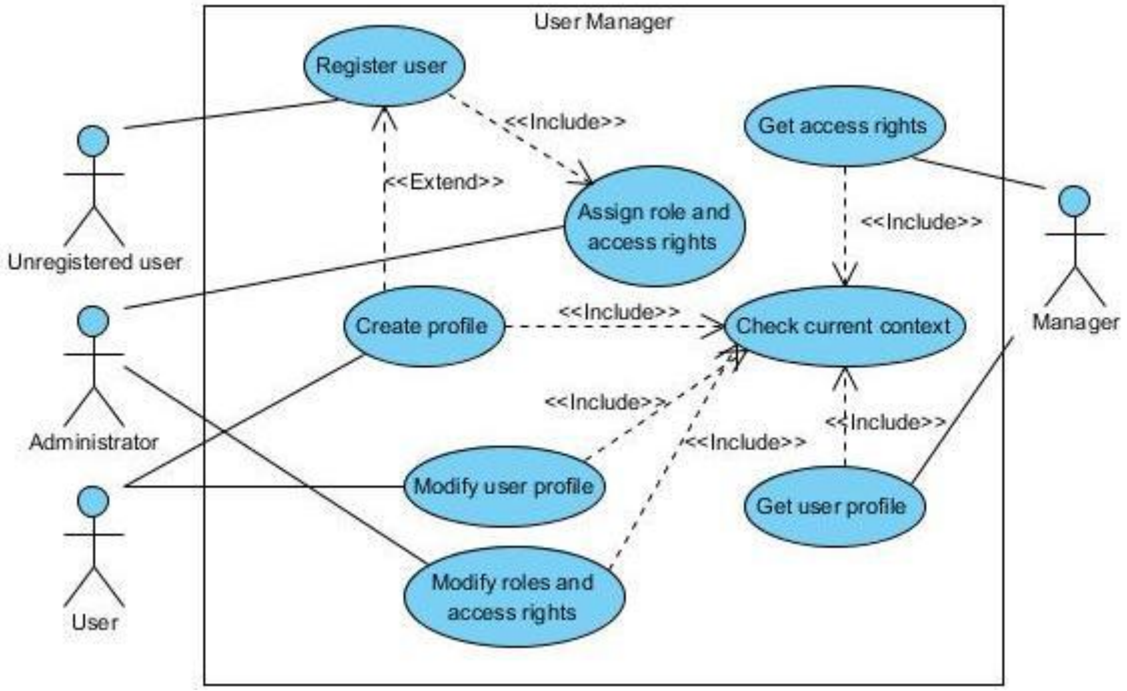


Figure 57 Use cases supported by User Manager

Another functionality provided by the component is user profile management. Every user is able to update its own profile. This leads to better customization and personalisation of his/her tools and user interfaces. User profile is then provided to other components in the same way as access rights information.

7.16.4. Functionality description

The component serves as a provider of access rights for actions within the system, whether the user can or cannot do some operation. All this information is stored and retrieved from this component data structure. The basic question (of any component) then – “Is it allowed for user X with some role Y to do action A in current context C?” This is the used format for access rights query from any component to this component (C is added automatically by connection to process management component) and it is manager’s basic service.

The second functionality is related to creation of new users (registration) and assignment of role and access rights by the administration user. After user receives confirmation he/she is able to work with the system on single sign-on basis using authentication service of the application server.

User is also able to manage its own profile with specific settings on tools and whole system, which are helpful in making it more suitable for him/her in order to achieve better customization and personalisation. User profile management (creation, update, removing - setup to default) is supported by this component and combines settings for more tools and applications, which are working together in the OCOPOMO platform.

7.16.5. Component API

Function	Description
Get access rights	*To obtain access rights about the authorization for actions within the components.* **Input**: action – an object that describe action for which decision about the access should be identified user – an identification of user role – an identification of user's role **Output**: decision – answer for this specific access right (according to current context and component)
Get user profile	*To obtain user profile in order to customize and personalise his/her view of the system.* **Input**: user – an identification of user **Output**: profile – object containing profile information of user (according to current context and component)

Table 63 User Manager API

7.17. VERSION MANAGER

7.17.1. Relevant user requirements

T-5	Content Management System (CMS) functionality
T-8	Opinion polling tool – participation of users in polls – one vote per person
T-9	Opinion polling tool – participation of users in polls – possibility to modify the answers provided (versioning)
I-11	Closing the scenario generation process / versioning
I-13	Control of Scenario Generation process phases
I-14	Maintaining of scenarios and rules within the ICT toolbox
I-26	Version control of process models and/or agent models

7.17.2. Context of the component

A context of the Version Manager is depicted in Figure 58. The manager is expected to communicate with the following managers:

- Concept manager – to support versioning of concepts in CCD
- Content manager – to support versioning of content repository elements
- Document manager – to version of saved documents
- Link manager – to support versioning functionality for links
- Process manager – to support versioning functionality for process steps and resources
- Rule manager – versioning of rules

- Simulation manager – versioning support for simulation models

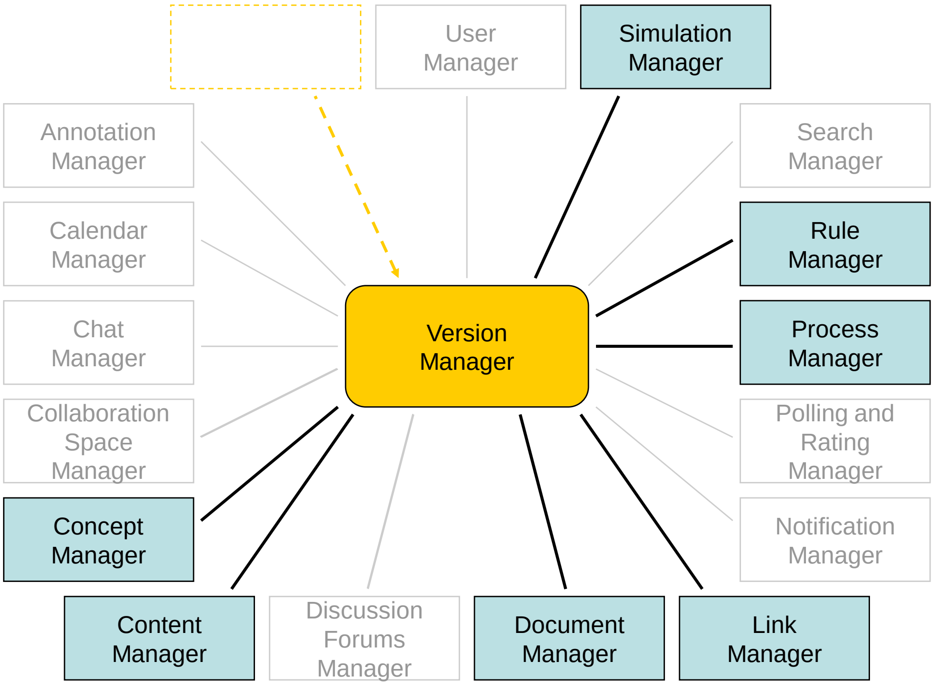


Figure 58 Context of Version Manager

7.17.3. Supported use cases

The aim of the component is to provide versioning functionality to other relevant components. In the presented use case diagram, Manager actor represents any manager that uses functionality of Version manager.

The component represents a closely coupled module, which provides versioning functionality for all relevant modules – Concept, Document, Link, Process, Rule and Simulation managers play the role of a user of this module.

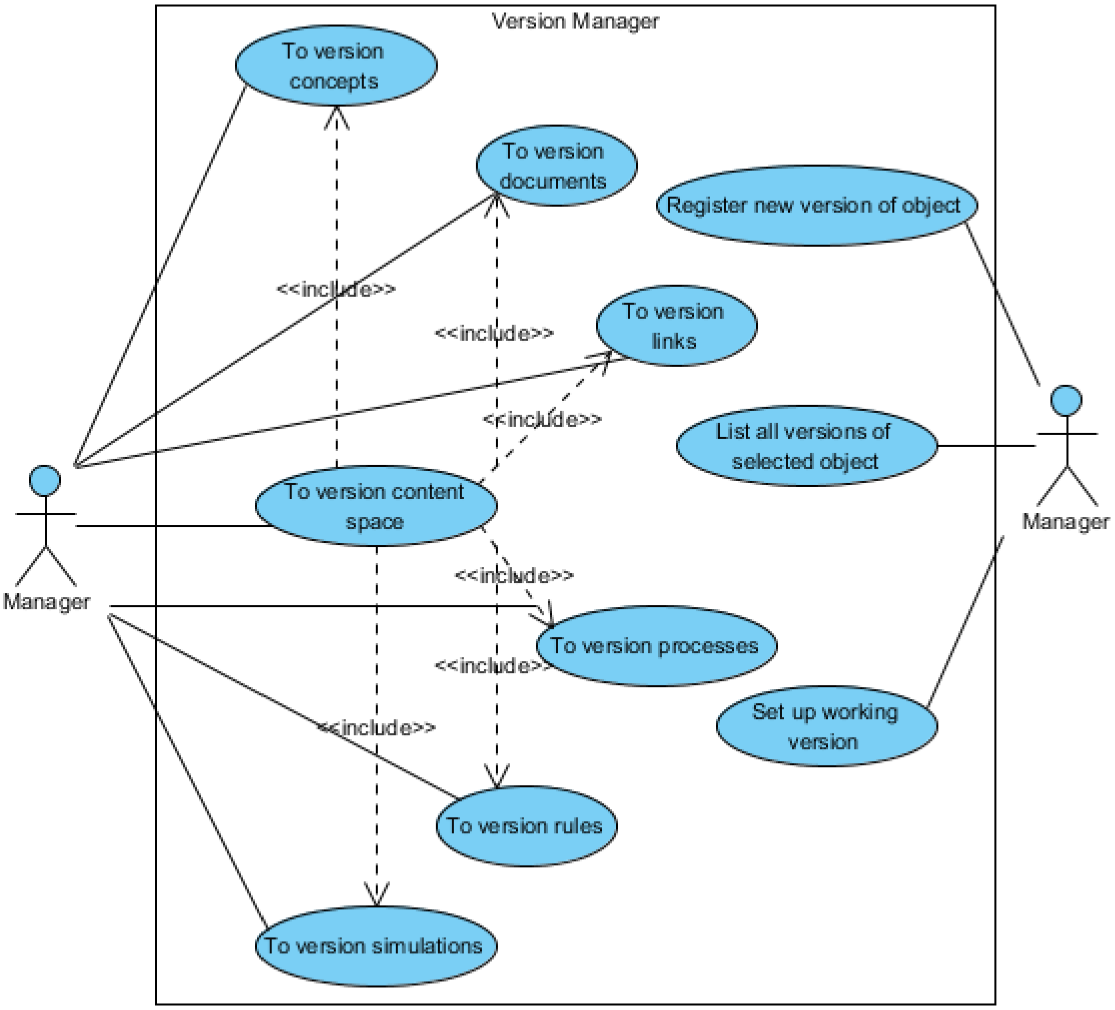


Figure 59 Use cases supported by Version Manager

7.17.4. Functionality description

Version manager is responsible for switching on and off versioning of various relevant objects either for individual objects or for all objects at once.

It is supposed to support versioning of several types of objects. Versioning allows storing several versions of some concept presented in Concept Manager as well as several versions of some document in Document Manager, workflow in Process Manager, rules/agents in Rule Manager and simulations in Simulation Manager. Several versions of links associated in the frame of the Link Manager are also available.

Switching versioning on/off in the frame of Content Manager means to switch versioning on/off for all objects stored by means of the Content Manager, as mentioned above.

Managers are provided with the possibility to register new version of a versioned object (content, document, workflow, simulation, rule, link), list all versions of a selected object and set up working (actual, current) version of an object – the version of the object (document, content, simulation etc.) which is used within the current session.

7.17.5. Component API

Function	Description
Switch object versioning on	*To switch versioning of an individual object on* **Input**: object ID **Output**: none
Switch object versioning off	*To switch versioning of an individual object off* **Input**: object ID **Output**: none
Switch space versioning on	*To switch versioning of the whole content space on* **Input**: space ID **Output**: none
Switch space versioning off	*To switch versioning of the whole content space off* **Input**: space ID **Output**: none
Register new version	*To register a new version of an object* **Input**: object **Output**: assigned version
List all versions of object	*To list all versions of an object present within the system* **Input**: object or object ID **Output**: list of available versions
Set up working version of object	*To set up a working version of an object* **Input**: object or object ID **Output**: none

Table 64 Version Manager API

8. ARCHITECTURE VALIDATION

In order to validate the architecture, the following procedure is conducted. For each requirement, which has been defined in this deliverable and in D1.1 [Bicking et al., 2010], it will be analysed if the proposed architecture is able to fulfil it. This is done by evaluating which of the components proposed is able to fulfil the requirement.

ID	Name	Priority	Component
I-NF-2	Accessibility	Must-have	
I-28	Action–based and rule-based role playing of stakeholders in simulation	Must-have	Simulation Manager
T-16	Agent-based simulation tool	Must-have	Simulation Manager
I-36	All personal preferences in one place	Nice-to-have	User Manager
I-NF-4	Authentication	Must-have	User Manager
I-NF-5	Authorization	Must-have	User Manager
T-37	Authorization/authentication issues are taken into account in individual tools	Must-have	User Manager
T-4	Chat	Must-have	Chat Manager
I-11	Closing the scenario generation process / versioning	Must-have	Process Manager, Version Manager
T-25	Commenting functionality	Must-have	Polling and Rating Manager
I-18	Comparison of simulations	Must-have	Simulation Manager
T-39	Computer-assisted Qualitative Data Analysis Software Tool – Coding of text passages and clustering of codes	Must-have	Annotation Manager, Concept Manager, Document Manager, Link Manager
T-40	Computer-assisted Qualitative Data Analysis Software Tool – flexible querying of codes and issues	Must-have	Annotation Manager, Concept Manager, Link Manager
T-41	Computer-assisted Qualitative Data Analysis Software Tool – statistics	Nice-to-have	Annotation Manager, Concept Manager, Link Manager
T-5	Content Management System (CMS) functionality	Must-have	Content Manager, Document Manager, Version Manager
I-13	Control of scenario generation process phases	Must-have	Process Manager, Version Manager
I-4	Creation of stakeholder groups for the scenario generation process	Must-have	Collaboration Space Manager, Notification Manager, Process Manager
I-23	Creation of stakeholders groups for policy modelling process	Must-have	Collaboration Space Manager, Notification Manager, Process Manager

I-22	Defining scenario for policy modelling	Must-have	Simulation Manager
I-17	Discussion about simulation results and decisions of human agents in simulation	Must-have	Discussion Forums Manager
T-1	Discussion forums	Must-have	Discussion Forums Manager
T-1-4	Discussion forums - Authorisation on level of the discussion forum	Must-have	Discussion Forums Manager, User Manager
T-1-5	Discussion forums - condition of use	Must-have	Discussion Forums Manager
T-1-2	Discussion forums - entries should be organised in threads	Must-have	Discussion Forums Manager
T-1-3	Discussion forums - possibility to order entries in chronological order and for topics	Must-have	Discussion Forums Manager
T-12	Discussion forums – moderated and non-moderated discussions	Must-have	Discussion Forums Manager
T-14	Discussion forums – rating of contributions and contributors (analysis of discussions based on a relevance feedback)	Must-have	Polling and Rating Manager
T-34	E-mail notification system	Must-have	Notification Manager
I-39	Full dependency graph including dependency of rules on lagged clauses	Must-have	Simulation Manager
I-NF-11	Help and assistance	Must-have	Notification Manager
T-C1	Hints for interesting topics	Nice-to-have	Annotation Manager, Document Manager, Notification Manager
I-29	Human actions analysis	Should-have	Simulation Manager
I-1	ICT toolbox functionality provided through one portal-based interface	Must-have	Collaboration Space Manager, Process Manager, Search Manager
T-18	Import of the previously exported simulation data	Should-have	Simulation Manager
I-5	Integration of components within the e-participation tools for scenario generation – data exchange / annotation	Must-have	Collaboration Space Manager, Concept Manager, Discussion Forums Manager, Link Manager, Search Manager
I-6	Integration of components within the e-participation tools for scenario generation – search	Must-have	Search Manager
I-7	Integration of components within the e-participation tools for scenario generation – workspace	Must-have	Collaboration Space Manager, Notification Manager, Process Manager,

			Search Manager
I-25	Integration of policy modelling tool and simulation / analysis tools – data exchange / annotation	Should-have	Concept Manager, Content Manager, Link Manager, Rule Manager
I-NF-10	Integrity	Must-have	
I-20	Log of activities within policy modelling / simulation	Must-have	Process Manager, Search Manager, Simulation Manager
I-19	Log of activities within scenario generation	Must-have	Annotation Manager, Process Manager, Search Manager
I-F-I3	Login	Must-have	User Manager
I-NFT-8	Look and feel	Must-have	
I-14	Maintaining of scenarios and rules within the ICT toolbox	Must-have	Concept Manager, Link Manager, Rule Manager, Version Manager
I-35	Multilingual interface	Must-have	
T-24	News functionality	Must-have	Notification Manager
T-C2	News – rating/polling functionality	Should-have	Polling and Rating Manager
T-29	Newsletter	Must-have	Notification Manager
I-NF-7	Operational	Must-have	
I-10	Opinion polling about the current version of scenario generation resources	Must-have	Polling and Rating Manager, Process Manager
T-10	Opinion polling tool – different types of questions & answers	Must-have	Polling and Rating Manager
T-7	Opinion polling tool – open forms	Must-have	Polling and Rating Manager
T-8	Opinion polling tool – participation of users in polls – one vote per person	Must-have	Polling and Rating Manager, Version Manager
T-9	Opinion polling tool – participation of users in polls – possibility to modify the answers provided (versioning)	Must-have	Polling and Rating Manager, Version Manager
T-11	Opinion polling tool – presentation of the results	Must-have	Polling and Rating Manager
I-F-I1	Password reminder	Must-have	User Manager
I-F-I6	Personalise overview	Must-have	Collaboration Space Manager, Notification Manager, Process Manager, Search Manager, User Manager
T-17	PM (Analysis) - Export of simulation-related data	Should-have	Simulation Manager
TP-3	PM (Analysis) - Narrative output	Must-have	Concept Manager, Link

			Manager, Simulation Manager
T-23	PM (Analysis) - Qualitative representation of the simulation results	Must-have	Concept Manager, Link Manager, Simulation Manager
TP-5	PM (Analysis) - Visualisations of non-numerical outcomes/events	Must-have	Simulation Manager
TP-2	PM (Analysis) – Experiment and rule development browser	Should-have	Concept Manager, Link Manager, Simulation Manager
TP-1	PM (Analysis) – Within-timestep dependency graph visualisation	Must-have	Concept Manager, Link Manager, Simulation Manager
FR26_PM	PM (Experimentation) - Automated experimentation	Must-have	Simulation Manager
FR25_PM	PM (Experimentation) - Change simulation parameters	Must-have	Simulation Manager
FR22_PM	PM (Experimentation) - User engagement in simulation	Must-have	Simulation Manager
FR23_PM	PM (Experimentation) - User interaction	Must-have	Simulation Manager
FR24_PM	PM (Experimentation/Gaming) - Gaming	Must-have	Simulation Manager
FR27_PM	PM (Gaming) - Feedback on simulation	Must-have	Simulation Manager
T-32	PM (Gaming) – Role-playing games (single user)	Must-have	Simulation Manager
T-33	PM (Gaming) – User interface for human player	Must-have	Simulation Manager
FR07_PM	PM (Modelling process) - Agent type creation	Must-have	Concept Manager, Link Manager, Rule Manager
FR08_PM	PM (Modelling process) - Agents at different aggregation levels	Must-have	Concept Manager, Link Manager, Rule Manager
NFR03_PM	PM (Modelling process) - End states	Must-have	Rule Manager
FR10_PM	PM (Modelling process) - Environment definition - general	Must-have	Concept Manager, Link Manager, Rule Manager
FR09_PM	PM (Modelling process) - Exogenous factors	Must-have	Concept Manager, Link Manager, Rule Manager
NFR07_PM	PM (Modelling process) - General model description	Must-have	Rule Manager
NFR04_PM	PM (Modelling process) - Initial model definition (Beginner's mode)	Should-have	Rule Manager
NFR05_PM	PM (Modelling process) - Iterations	Should-have	Rule Manager

	(Expert's mode)		
NFR06_PM	PM (Modelling process) - Model description	Must-have	Rule Manager
FR13_PM	PM (Simulation setup) - Initial state definition	Must-have	Simulation Manager
FR12_PM	PM (Simulation setup) - Setup initial agent facts	Must-have	Simulation Manager
FR11_PM	PM (Simulation setup) - Setup world facts	Must-have	Simulation Manager
FR17_PM	PM (Simulation termination) - Adjustable parameters	Must-have	Simulation Manager
FR14_PM	PM (Simulation termination) - End state	Must-have	Simulation Manager
FR15_PM	PM (Simulation termination) - Irregular termination events	Must-have	Simulation Manager
FR16_PM	PM (Simulation termination) - Regular termination events	Must-have	Simulation Manager
FR21_PM	PM (Simulation termination) - Simulation abort	Must-have	Simulation Manager
FR20_PM	PM (Simulation termination) - Simulation interrupt	Must-have	Simulation Manager
FR19_PM	PM (Simulation termination) - Simulation start	Must-have	Simulation Manager
FR18_PM	PM (Simulation termination) - State validation	Must-have	Simulation Manager
NFR08_PM	PM (Simulation) - Event handling	Must-have	Simulation Manager
NFR09_PM	PM (Simulation) - Exception handling	Must-have	Simulation Manager
NFR12_PM	PM (Simulation) - Parameter locking	Must-have	Simulation Manager
NFR11_PM	PM (Simulation) - Parameter presentation	Must-have	Simulation Manager
NFR14_PM	PM (Simulation) - Simulation execution	Must-have	Simulation Manager
NFR10_PM	PM (Simulation) - Simulation visualization	Must-have	Simulation Manager
NFR13_PM	PM (Simulation) - State handling for inspection	Must-have	Simulation Manager
FR06_PM	PM (Transformation process) - Assumption definition	Must-have	Concept Manager, Link Manager, Rule Manager
NFR01_PM	PM (Transformation process) - Data representation	Must-have	Concept Manager, Link Manager
FR01_PM	PM (Transformation process) - Define initial policy modelling aspects	Must-have	Concept Manager, Link Manager, Rule Manager

FR03_PM	PM (Transformation process) - Environment generation	Must-have	Concept Manager, Link Manager, Rule Manager
FR04_PM	PM (Transformation process) - Goal definition	Must-have	Concept Manager, Link Manager, Rule Manager
NFR02_PM	PM (Transformation process) - Language transition	Should-have	Annotation Manager, Rule Manager
FR05_PM	PM (Transformation process) - Rule generation	Must-have	Concept Manager, Link Manager, Rule Manager
FR02_PM	PM (Transformation process) - Stakeholder extraction	Must-have	Concept Manager, Link Manager, Rule Manager
T-22	Preview simulation mode – focusing on a part of the used model	Must-have	Simulation Manager
T-20	Preview simulation mode – level of details and/or time scale	Must-have	Simulation Manager
T-21	Preview simulation mode – searching for a specified event	Should-have	Simulation Manager
T-19	Previewing of a simulation (means: state of running simulation can be observed)	Must-have	Simulation Manager
I-NF-6	Privacy	Must-have	User Manager
I-24	Publishing of simulation results by the publishing tool (content management tool)	Must-have	Document Manager
I-F-I2	Removing profile	Must-have	User Manager
I-NF-3	Response Time	Must-have	
T-30	RSS	Must-have	Notification Manager
T-28	Shared calendar with events related to the current processes	Should-have	Calendar Manager, Notification Manager
I-34	Simulation back-end integrated with the ICT toolbox	Must-have	Simulation Manager
I-27	Simulation preview tool available from different physical locations – remote access	Must-have	Simulation Manager
T-36	Simulation tool – performance in simulation cycles	Must-have	
T-35	Simulation tool – the number of agents	Must-have	
I-3	Starting the scenario generation process - initial scenario	Must-have	Process Manager
I-12	Support for direct export/import of information between scenario generation process and policy modelling	Should-have	Concept Manager, Link Manager, Process Manager, Rule Manager

I-15	Support for the policy modelling tool to create a new scenario generation iteration	Must-have	Process Manager
T-42	Tags	Should-have	Calendar Manager, Document Manager, Discussion Forums Manager, Polling and Rating Manager
T-43	Tags - automatic support	Nice-to-have	
T-6	Teleconferencing tool	Nice-to-have	
T-38	Transcription tool	Should-have	
I-2	Transformation table - connection of context-specific information within the Scenario Generation and Policy Modelling process in ICT toolbox	Must-have	Annotation Manager, Collaboration Space Manager, Concept Manager, Link Manager, Process Manager
I-40	Transition table browser	Must-have	Concept Manager, Link Manager, Rule Manager
I-30	Translation of agent rules from a tool neutral syntax into simulation back-end language	Must-have	Rule Manager
I-NF-1	Usability	Must-have	
I-F-I5	User profile	Must-have	Collaboration Space Manager, Notification Manager, User Manager
I-F-I4	User registration	Must-have	User Manager
I-26	Version control of process models and/or agent models	Must-have	Version Manager
I-32	Workflow support	Must-have	Collaboration Space Manager, Notification Manager, Process Manager
	New requirements		
SOTA-2	Content/WYSIWYG	Should-have	Document Manager
UC-10	Development of social network	Should-have	Link Manager, Rule Manager
UC-7	Expertise-based relations	Should-have	Annotation Manager
SOTA-3	File types supported	Should-have	Document Manager
UC-6	Generation of relations	Should-have	Annotation Manager
UC-4	Initiate project	Must-have	Collaboration Space Manager, Process Manager
SOTA-6	Information structuring	Should-have	Annotation Manager, Rule Manager
UC-2	Invitation – send and receive	Should-have	Notification Manager, Process Manager, User

			Manager
SOTA-7	Memos	Should-have	Annotation Manager, Document Manager
UC-9	Network visualisation	Should-have	Rule Manager
SOTA-8	Non-RETE rule engine	Must-have	Simulation Manager
UC-8	Quantitative data analysis	Should-have	Annotation Manager
UC-1	Rights management	Must-have	Collaboration Space Manager, Process Manager, User Manager
UC-3	Send request for invitation	Nice-to-have	Process Manager, User Manager
SOTA-4	Several document editors	Should-have	Document Manager
SOTA-5	Real-time co-editing	Nice-to-have	Document Manager
UC-5	Update description of the project	Must-have	Collaboration Space Manager, Document Manager
SOTA-1	Workflow engine	Should-have	Document Manager, Process Manager

Table 65 Requirement coverage

A few requirements are not covered by the proposed architecture (marked in yellow and reddish colours). The yellow colour represents a few mainly non-functional requirements, some of which have been discussed within architectural perspectives but are not covered by particular components (e.g. I-NF-2 Accessibility, I-NF-10 Integrity, I-NFT-8 Look and feel, I-NF-7 Operational, I-35 Multilingual interface, I-NF-1 Usability). These requirements represent features which must be taken into consideration during later implementation phases of the project and therefore it is not possible to judge on their satisfaction now.

Similarly, there is a set of yellow requirements on performance characteristics of the prospective OCOPOMO ICT toolbox (e.g. T-36 Simulation tool - performance in simulation cycles, I-NF-3 Response time, T-35 Simulation tool – the number of agents). These characteristics have quantitative nature and they can be evaluated only after the implementation will be ready.

Only three requirements (marked in reddish colour) are not covered by the proposed architecture – T-6 Teleconferencing tool, T-38 Transcription tool, and T-43 Tags - automatic support. Two of them have ‘Nice-to-have’ priority and one has ‘Should-have’ priority. This assignment means that the actual decision whether to accept or reject such requirements should be based on available resources [Bicking et al., 2010]. We have decided not to consider these requirements yet and to postpone the decision on possible incorporation into the ICT toolbox to later phases of the project. The reason is that, on one hand, they represent functionality the absence of which has no implication on the ability and functioning of the toolbox to support users in their tasks (currently they do not play any role in the processes defined by the OCOPOMO approach) and, on the other hand, their incorporation would be quite costly in terms of necessary resources.

More detailed validation of the proposed architecture and its breakdown into managers is expected to be performed within workpackage WP3, especially considering (in detail) functionality provided by the selected software tools to be reused.

In order to illustrate, how the proposed architecture can be used to support user activities, we present an example of a user scenario to illustrate the topic. This is a simple example of a scenario flow which includes these operations of one user (Facilitator)[132]:

- Facilitator opens the document for analysis (he/she already knows which document).
- Then he/she highlights some text in order to create an issue (a concept object).
- Then he/she inserts a relation to another issue (concept) and links them together.

The model of the use case scenario (as a sequence diagram) is presented in Figure 60. It involves 7 managers. Communication between these managers (as well as a user - Facilitator) consists of the following steps:

1. Facilitator decides to analyse the scenario document using Annotation Manager (starts with opening of this document). A call to the manager is performed in order to achieve this goal using his/her user interface.
2. Annotation Manager calls Process Manager in order to obtain context information (details regarding the current process - project).
3. Context information is returned to Annotation Manager and is available from this moment within the current session of user.
4. Annotation Manager calls User Manager for access rights of the current user (if Facilitator is able to perform such analysis).
5. Access rights information about the current operation (opening the scenario) is returned to Annotation Manager.
6. Document Manager is called for getting the document (for analysis).
7. Document Manager wants to retrieve anything necessary (document, metadata, etc.) from Content Manager.
8. All data related to document (for analysis) is returned to Document Manager from Content Manager.
9. Document with all its details (e.g. metadata) is returned to Annotation Manager and is ready for the analysis.
10. Annotation Manager provides all its features with opened document to Facilitator.
11. Facilitator wants to create a concept object (annotation based on the highlighted text and its metadata). He/she uses interface on the screen for highlighting and inserting metadata of the new concept and clicks for an action. After this moment Annotation Manager has necessary data for creation of the new concept object element.
12. Annotation Manager calls User Manager for retrieving access rights regarding the current operation (creation of concept object).
13. Access rights are retrieved from the User Manager.
14. Annotation Manager calls Concept Manager and sends all necessary data for creation of a concept object.
15. Concept Manager uses Content Manager for storage of the new concept object using its API.

[132] We assume that the user is already authenticated and he/she is currently working with the annotation tool.

16. Concept Manager retrieves identification of new the concept object, which was stored in respective content repository.

17. Annotation Manager retrieves identification of the new concept which was created according to its needs.

18. Facilitator is able to see the new concept in his/her user interface.

19. Facilitator now wants to create a link between the new concept and some other issue (also concept object). He/she selects the second concept object, new one and adds metadata related to the new link, which will be created between them. Then he/she clicks for action and after this moment Annotation Manager has all necessary data for the creation of the link.

20. Annotation Manager calls User Manager for access rights regarding the current operation (creation of link object).

21. Access rights are retrieved from the User Manager.

22. Annotation Manager needs to retrieve the second concept object and its details, therefore Concept Manager is called here.

23. Concept Manager uses Content Manager for retrieving the concept object using its API.

24. Concept Manager retrieves the concept object from the content repository.

25. Annotation Manager retrieves the concept object from Concept Manager.

26. Annotation Manager now has both concept objects (the new object created before and retrieved second concept) and all necessary data for the new link. Now it calls API of Link Manager in order to create the new link object.

27. Link Manager uses Content Manager to store the new link object using its API.

28. Link Manager retrieves identification of the new link which was stored in the content repository.

29. Annotation Manager retrieves the identification of the new concept which was created according to its needs.

30. Facilitator is able to see the new link in his/her user interface.

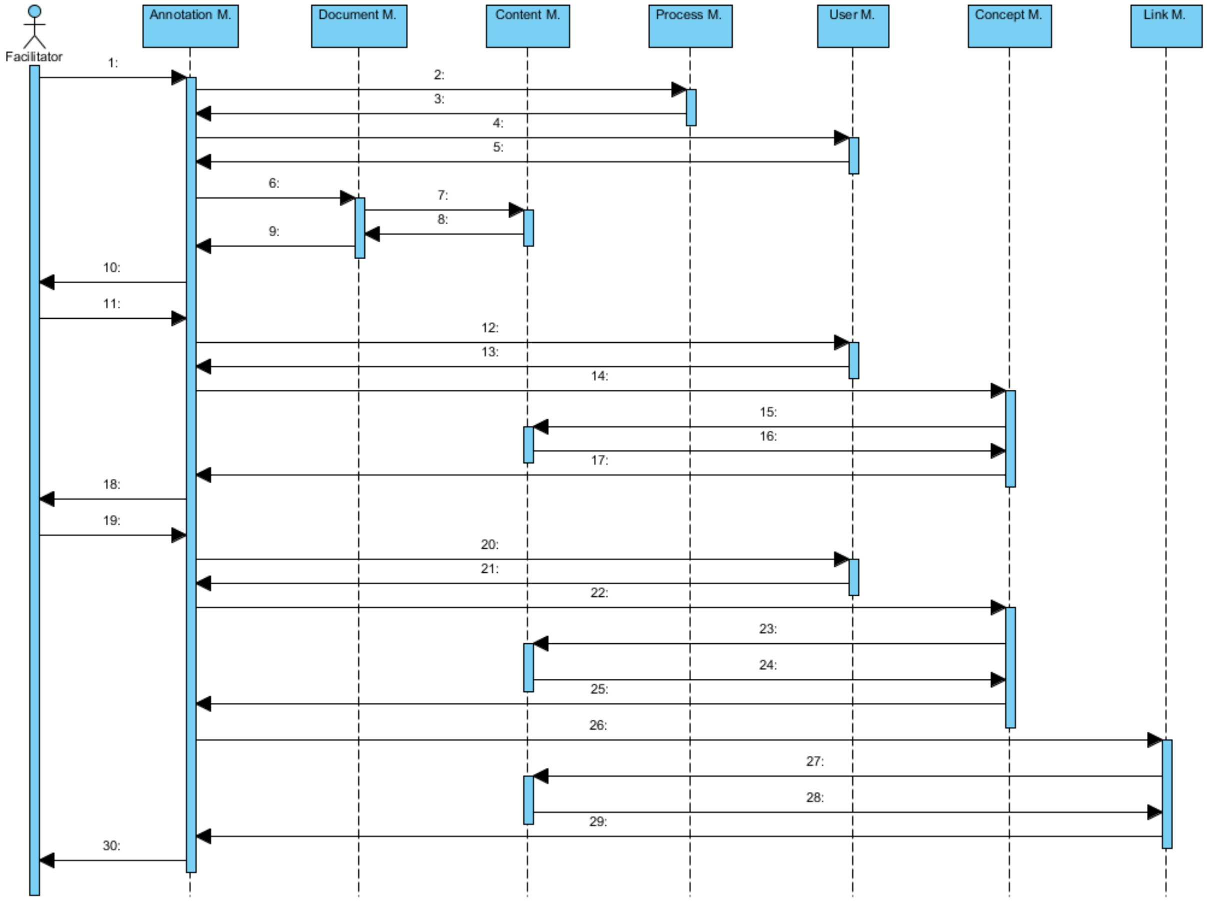


Figure 60 Sequence diagram for a simple use case scenario 1.

9. CONCLUSION

In this document we have outlined an overall architecture and defined all necessary components that are considered to be essential for the prospective OCOPOMO ICT toolkit. All the activities related to the architecture design and all decisions taken within these activities can be loosely divided into the following three areas:

- Initial understanding of the prospective system
- State of the art analysis
- Architecture and component design

Initial understanding of the prospective system corresponds to the current[133] level of understanding of the processes hidden behind collaborative preparation of evidence-based scenarios, the transformation of evidence-based scenarios into formal simulation models, and utilising feedback from simulation-based scenarios back to evidence-based scenarios. This section is based on information available from the previous workpackage WP1 [Bicking et al., 2010] and the ongoing parallel workpackage WP5 [Moss et al., 2010]. This area comprises the definition of system boundaries and user-oriented perspective of the system and supported processes. The main objective was to complete the vision and understanding of the platform.

State of the art analysis was the next area of interest. The main objective is to provide potential solutions for particular issues in platform architecture with an emphasis on current needs. Therefore, we have investigated current technologies and software tools in areas identified as relevant based on initial understanding of the prospective system (i.e. e-participation, scenario generation and analysis, formal modelling, integration, relevant standards). For all of them we have identified several alternatives for software solutions and criteria for their comparison and evaluation. According to all current project needs, user requirements, implementation considerations and their combinations, three software tools have been selected to be reused.

To describe OCOPOMO architecture we have adopted a well-known methodology standardized under IEEE 1471. All architectural aspects were identified through architectural views and perspectives. Two basic views were used - Functional view (structure of the platform and basic description of components) and Information view (data model). Also three perspectives were considered in order to refine the architecture – Interaction perspective (issues related to GUI), Usability perspective (enabling users to utilise the platform effectively), and Internationalisation perspective (support for different languages). According to current needs and decisions the architecture was developed as a set of managers. Next, all the components (managers) were described in more details according to their functional behaviour, supported use cases (manager-specific use cases) and APIs available to other components.

State of the art evaluation has resulted in the selection of these software tools:

- Alfresco CMS/Share - basic CMS-based tool with personalised user interface for collaboration and document management features (available by Alfresco Share application), important also as an integration platform for content repository (CMIS standard supported) and presentation integration (portal solution based on Share), all within web/application server with a database (current preferences: Tomcat, MySQL)

[133] This understanding can evolve in next project phases and therefore the presented design (e.g. use cases, mock-ups, etc.) cannot be considered frozen and unchangeable but rather flexible - able to evolve and accommodate future required modifications.

- Simulation software – an important combination of agent-based simulation software (Repast will be reused as a general ABM platform) and rule engine (DRAMS, currently under development) for creation and running of simulation models.

Based on the decision on selection of existing tools and their reuse, some of the managers should be implemented from the scratch while others will be re-used or adapted for usage in the OCOPOMO platform from existing tools.

Figure 61 represents an updated version of the overall architecture – as a combination of the proposed architecture and the selection of tools to be reused within the ICT toolkit. The architecture components (managers) are shown in different colours according to the way they are expected to be implemented.

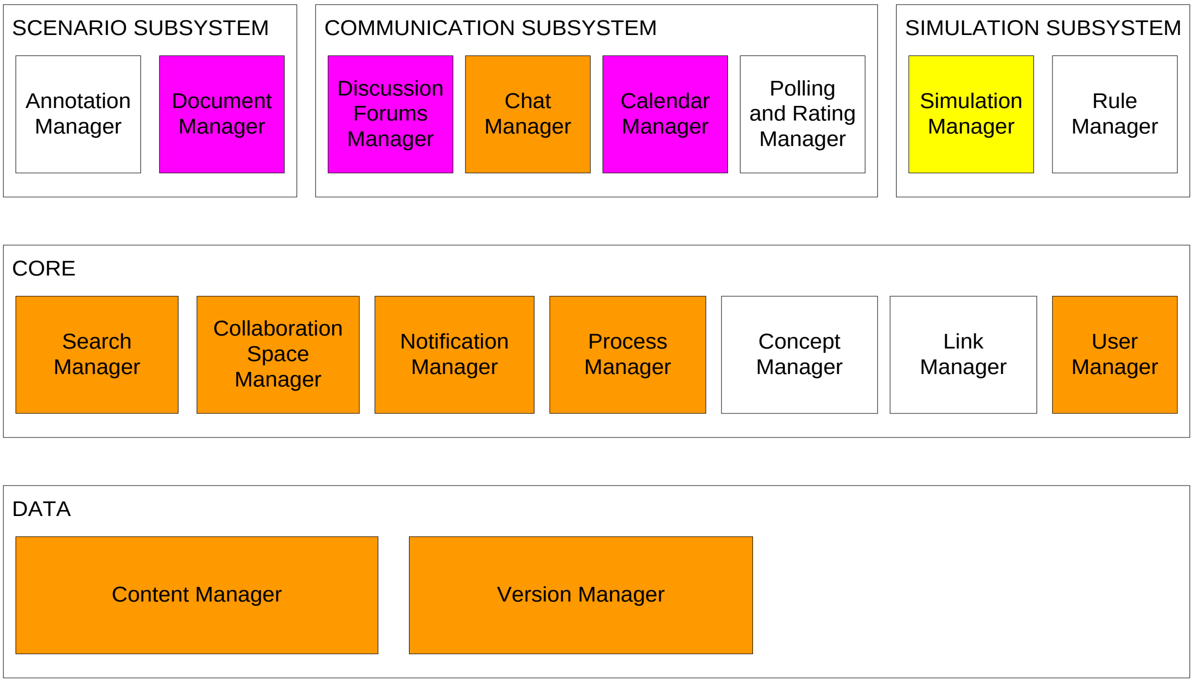


Figure 61 Platform components - implementation needs for managers

Managers depicted in white colour will be implemented from scratch as platform specific components. Here we have Concept, Link, Polling and Rating, Rule and Annotation managers. These managers (except Polling and Rating manager) are very specific to identified process and its needs (scenario analysis, creation and maintenance of concepts and linking objects, creation and maintenance of rules, agents, models, etc.) and therefore we were not able to find any tool to provide the required functionality – the only possibility is to implement their functionality. Although Polling and rating manager is expected to provide rather standard functionality, since it is missing in the Alfresco CMS, it must be added (probably by adapting an existing code).

There is one manager with yellow background – Simulation manager. It combines Repast (as reused software for simulations) and DRAMS (rule engine which works upon Repast's simulation infrastructure to support declarative agent modelling). As it was already written, DRAMS is under active development of project partners responsible for formal modelling. Therefore, this manager is partially being developed from the scratch and partially reusing existing software – the DRAMS part must be developed further to obtain all the required functionality of the manager.

Managers with magenta background can be directly replaced with the functionality provided by the selected Alfresco software (maybe they will be only customised or only small implementation will be needed). All of them (Document, Discussion forums and Calendar managers) are implemented within Alfresco Share application.

All the other (orange background) managers are those which are partially supported by the selected Alfresco, but it is expected that some implementation effort will be necessary for integration of them within OCOPOMO platform and for extension of their functionality.

All implementation work on platform components will be controlled and also done within the workpackage WP3 and its particular tasks. Task 3.1 is expected to test the selected software tools and their limits according to project’s needs in detail. Especially those managers (in magenta and orange) which are going to reuse some tool from the Alfresco CMS/Share should be analysed and any bottlenecks for additional implementation of expected features (or their customisation) should be identified. The identified missing functionality in managers with magenta and/or orange background as well as functionality of managers with white background must be implemented. The main task for this will be Task 3.2 - Implementation of platform components, which is then iterated after the first trial by revision of implementation according to feedback from testing (Task 3.3). Integration of platform and its finalization will be controlled by the integration workpackage WP4.

10. REFERENCES

OCOPOMO documents

[Bicking et al., 2010] Bicking, M., Butka, P., Delrio, C., Dunilova, V., Hilovska, K., Kacprzyk, M., Lotzmann, U., Lucznik, K., Mach, M., Moss, S., Nowak, A., Pizzo, C., Rinaldi, V., Roszczynska-Kurasinska, M., Sabol, T., Scherer, S., Schmidt, A., Ventzke, S. and Wimmer, M. A.: D1.1 Stakeholder Identification and requirements for toolbox, scenario process and policy modelling. Deliverable 1.1, OCOPOMO, 2010.

[Moss et al., 2010] Moss, S. et al.: D5.1 Scenario, policy model and rule-based agent design. Deliverable 5.1, OCOPOMO, 2010 (under preparation).

[Ocopomo-DoW, 2009] OCOPOMO: FP7-ICT-2009.7.3 Grant Agreement for Collaborative Project, Annex I: “Description of Work”, Grant agreement number 248128. Date of approval: 2009-12-14

Integration methods and technologies

[Cerami, 2002] Cerami, E.: Web Services Essentials. O'Reilly Media, Inc., 2002, 304p.

[Cover, 2004] Cover, R.: Java Business Integration (JBI) Specification Early Draft Review. The XML Cover Pages 2004, http://xml.coverpages.org/JBI-EarlyReview.html [last accessed 20.7. 2010].

[Fielding, 2000] Fielding, R.: Representational state transfer (REST). Chapter 5 in Architectural Styles and the Design of Network based Software Architectures, 2000.

[Gudgin et al., 2007] Gudgin, M., Hadley, M., Mendelsohn, N., Moreau, J. and Nielsen, H.: Simple Object Access Protocol (SOAP) 1.2, World Wide Web Consortium, 2007.

[High et al., 2005] High, R., Kinder, S. and Graham, S.: IBM’s SOA Foundation – An Architectural Introduction and Overview. IBM, 2005, http://download.boulder.ibm.com/ibmdl/pub/software/dw/webservices/ws-soa-whitepaper.pdf [last accessed 20.7.2010].

[Hohpe and Woolf, 2003] Hohpe, G. and Woolf, B.: Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Addison-Wesley Professional, Boston, USA, 2003, 736p.

[Hollingsworth, 1995] Hollingsworth, D.: Workflow management Coalition - The Workflow Reference Model. WfMC-TC-1003, 1995.

[Juric et al., 2002] Juric, M., Nashi, N., Berry, C., Kunnumpurath, M., Carnell, J. and Romanosky, S.: J2EE Design Patterns Applied. Wrox, UK, 2002, 450p.

[Juric et al., 2007] Juric, M., Loganathan, R., Sarang, P. and Jennings, F.: SOA Approach to Integration. Packt Publishing, Birmingham, UK, 2007, 384p.

[Kinnumpurath, 2005] Kinnumpurath, M.: JBI – A Standard-based approach for SOA in Java. TheServerSide, 2005, http://www.theserverside.com/tt/articles/article.tss?l=JBIforSOA [last accessed 20.7.2010].

[La et al., 2007] La, H.Y., Bae, J.S., Chang, S.H. and Kim, S.D.: Practical Methods for Adapting Services Using Enterprise Service Bus. In: Proceedings of the 7th International Web Engineering Conference ICWE 2007, Springer LNCS 4607, Como, Italy, 53-58.

[McLean et al., 2002] McLean, S., Naftel, J. and Williams, K.: Microsoft .NET Remoting. Microsoft Press, 2002.

[Ouyang, 2006] Ouyang, Ch., Dumas, M., van der Aalst, W.M.P. and Hofstede, A.H.M: From Business Process Models to Process-oriented Software Systems: The BPMN to BPEL Way.

Technical Report, Queensland University of Technology, 2006, http://eprints.qut.edu.au/5266/1/5266.pdf [last accessed 20.7.2010].

[Papazoglou, 2003] Papazoglou, M.: Service-oriented computing: concepts, characteristics and directions. In: Proceedings of the Fourth International Conference on Web Information Systems Engineering, 2003.

[Ten-Hove, 2006] Ten-Hove, R.: JBI Components: Part 1 (Theory). Sun Microsystems, Inc., 2006, https://open-esb.dev.java.net/public/pdf/JBI-Components-Theory.pdf [last accessed 20.7.2010].

[Vinoski, 2006] Vinoski, S.: Advanced Message Queuing Protocol. IEEE Internet Computing 10, 2006, 87–89.

E-participation tools and technologies

[Albrecht et al., 2008] Albrecht, S., Kohlrausch, N., Kubicek, H., Lippa, B., Märker, O., Trénel, M., et al.: eParticipation - Electronic Participation of Citizens and the Business Community in eGovernment. Bremen: Study on Behalf of the Federal Ministry of the Interior (Germany), Division IT 1, 2008.

[Aichholzer and Allhutter, 2009] Aichholzer, G. and Allhutter, D.: Public Policies on e-participation in Austria. In: Macintosh, A. & Tambouris, E. (Eds.): Electronic Participation: First International Conference, ePart 2009, Linz, Austria, September 2009, Proceedings (LNCS), Berlin, Heidelberg, 24-45.

[Bicking and Wimmer, 2009] Bicking, M. and Wimmer, M. A.: Evaluation Framework to Assess e-participation Projects in Europe. In: Tambouris, E., Macintosh, A. (Eds.): Electronic Participation: Proceedings of Ongoing Research, General Development Issues and Projects of ePart 2009, Linz, Trauner Verlag, 2009, 73-82.

[Charalabidis et al., 2009] Charalabidis, Y., Koussouris, S. and Kipenis, L.: Report on the Objectives, Structure and Status of e-participation Initiative Projects in the European Union. Whitepaper, MOMENTUM Konsortium, 2009, http://www.ep-momentum.eu/ [last accessed 22.09.2010].

[cms, 2009] Water&stone and CMSWire: Open Source CMS Market Share. Technical report, 2009, http://www.waterandstone.com/2009-open-source-cms-market-share-report-august-2009 [last accessed: 22.09.2010].

[Koulolias et al., 2006] Koulolias, V., Karamagioli, E. and Xenakis, A.: The Gov2DemOSS eParticipation Platform: A New Era Tool for eDemocracy Implementation. In P. Cunningham and M. Cunningham (Eds.): Exploiting the Knowledge Economy: Issues, Applications, Case Studies, Proceedings of eChallenges 2006, Amsterdam, 2006. IOS Press.

[Mintert, 2010] Mintert, S.: Marktübersicht Web-CMS Unteilbares. iX Magazin für professionelle Informationstechnik, 2010 (8), 104–109.

[oecd, 2007] Participative Web and User-Created Content: Web 2.0, Wikis and Social Networking. OECD, 9, 2007.

[Panopoulou et al., 2009] Panopoulou, E., Tambouris, E. and Tarabanis, K.: E-participation initiatives: How is Europe progressing? European Journal of ePractice, 7, 2009.

[Scherer et al., 2008] Scherer, S., Schneider, C. and Wimmer, M. A.: Studying e-participation in Government Innovation Programmes: Lessons from a Survey. In: Hampe, F., Swatman, P. M., Gricar, J., Pucihar, A. and Lenart, G. (Eds.): eCollaboration: Overcoming Boundaries through Multi-Channel Interaction. 21st Bled eConference, 2008, electronic proceedings.

[Scherer et al., 2009a] Scherer, S., Holzner, M., Karamagioli, E., Lorenz, M., Schepers, J. and Wimmer, M. A.: Integrating Semantics, Social Software and Serious Games into e-participation: The VoiceS Project. In: Electronic Participation: Proceedings of Ongoing

Research, General Development Issues and Projects of ePart 2009, 1st International Conference, ePart 2009 Linz, Austria, September 1-3, 2009, Linz, Austria, Trauner, 151-158.

[Scherer et al., 2009b] Scherer, S., Karamagioli, E., Titorencu, M., Schepers, J., Wimmer, M. A. and Koulolias, V.: Usability Engineering in e-participation. European Journal of ePractice, 7, 2009.

[Scherer et al., 2011] Scherer, S., Liotas, N., Wimmer, M. A., Tambouris, E. and Tarabanis, K.: Interoperability Requirements, Recommendations and Standards in E-Participation. In: Charalabidis, Y. (Ed.): Interoperability in Digital Public Services and Administration: Bridging E-Government and E-Business, chapter 6, will be published in 2011, IGI Global. ISBN 978-1-61520-887-6.

[Schwartz et al., 2004] Schwartz, L., Clark, S., Cossarin, M. and Rudolph, J.: Educational Wikis: features and selection criteria, The International Review of Research in Open and Distance Learning, 2004, http://www.irrodl.org/index.php/irrodl/article/view/163/244 [last accessed: 06.09.2010].

[Tambouris et al., 2007] Tambouris, E., Liotas, N. and Tarabanis, K.: A Framework for Assessing e-participation Projects and Tools. In: HICSS ’07: Proceedings of the 40th Annual Hawaii International Conference on System Sciences 2007, pp. 90, Washington, DC. IEEE Computer Society.

[Tambouris et al., 2008] Tambouris, E., Kalampokis, E. and Tarabanis, K.: A survey of e-participation research projects in the European Union. International Journal of Electronic Business (IJEB), 6(6), 2008.

[Thorleifsdottir and Wimmer, 2006] Thorleifsdottir, A. and Wimmer, M. A.(Eds.): DEMO-net Deliverable 5.1: Report on current ICTs to enable Participation. DEMO-net Deliverable, 2006.

[Ventzke et al., 2010] Ventzke, S., Weiß, S. and Wimmer, M. A.: Stakeholder-Beteiligung in der Entwicklung eines Virtual Company Dossiers mithilfe von Web 2.0. In: Wimmer, M. A., Brinkhoff, U., Kaiser, S., Lück-Schneider, D., Schweighofer, E. and Wiebe, A. (Eds.): Vernetzte IT für einen effektiven Staat: Gemeinsame Fachtagung Verwaltungsinformatik (FTVI) und Fachtagung Rechtsinformatik (FTRI) 2010, number 162 in Lecture Notes in Informatics, 109–122.

[Westholm and Wimmer, 2007] Westholm, H. and Wimmer, M. A. (Eds.) DEMO-net Deliverable 6.2: Interdisciplinary framework to address the socio technical and political challenges of e-participation. DEMO-net Deliverable, 2007.

[Wimmer, 2007] Wimmer, M. A.: Ontology for an e-participation virtual resource centre. In: Janowski, T. and Pardo, T. A. (Eds.): ICEGOV ’07: Proceedings of the 1st international conference on Theory and practice of electronic governance, 2007, New York, 89-98.

Scenario generation and analysis tools and technologies

[Alexa and Zuell, 1999] Alexa, M. and Zuell, C.: A review of software for text analysis. ZUMA Nachrichten Spezial 5. ZUMA: Mannheim, 1999.

[Barry, 1998] Barry, C.A.: Choosing Qualitative Data Analysis Software: Atlas/ti and Nudist Compared. In: Social Research On-line, Vol. 3, 1998, Issue: 3, 1-17.

[Bertil and Magnusson, 2002] Bertil, R. and Magnusson, C.: Developing the art of argumentation: A software approach. In: Proceedings of the Fifth Conference of the International Society for the Study of Argumentation, 2002.

[Bicking and Wimmer, 2010] Bicking, M. and Wimmer, M.A.: Need for Computer-Assisted Qualitative Data Analysis in the Strategic Planning of E-Government Research. In: Chun, Soon Ae; Sandoval, Rodrigo; Philpot, Andrew: Public Administration On-line: Challenges and Opportunities. omni press, 2010.

[Bradfield et al., 2005] Bradfield, R., Wright, G., Burt, G., Cairns, G. and K. van der Heijden: The origins and evolution of scenario techniques in long range business planning. In: Futures, Vol. 37, 2005, Issue 8, 795-812.

[Brandao, 2006] Brandão Moniz, A.: Scenario-Building Methods as a Tool for Policy Analysis. In Innovative Comparative Methods for Policy Analysis, Beyond the Quantitative-Qualitative Divide, R. Rihoux, H. Grimm, Eds. Springer US, 2006, 185-209

[Creswell and Maietta, 2002] Creswell, J. and Maietta, R.: Qualitative Data Analysis Software. In: Miller, D.C., Salkind, N. J. (Eds) Handbook of Research Design and Social Measurement, 6th Edition, Thousand Oaks, CA: SAGE, 2002.

[Dilek, 2009] Dilek, C.: The Scenario Analysis Tool Suite: A User's Guide, 2009.

[Gausemeier et al., 1995] Gausemeier, J., Fink, A. and Schlake, O., Szenario-Management: Planen und Führen mit Szenarien. München, Hanser, 1995

[Geschka and Hammer, 1997] Geschka, H. And Hammer, R.: Die Szenario-Technik in der strategischen Unternehmensplanung. In Hahn, D.; Taylor, B. (eds.): Strategische Unternehmensplanung. 7th Edition, Heidelberg 1997, p. 464-489

[Gibson, 1996] Gibson, R.: Rethinking the futures. Nicholas Brealey Publishing, London, UK, 1996.

[Glenn et al., 1999] Glenn, J. (ed.), Futures research methodology. Washington: American Council for the United Nations University, CD Rom: version 1.0, 1999

[Jansen et al., 2006] Janssen, M. et al.: Deliverable D 2.1: Scenarios report (including regional workshops report), 2006, http://www.egovrtd2020.org/EGOVRTD2020/navigation/work_packages/wp2_scenario_building/deliverable_scenario_building

[Johnson et al., 2002] Johnson, G., Scholes, K. and Whittingon, R.: Exploring Corporate Strategy. Prentice Hall, 2002

[Kahn and Weiner, 1967] Kahn, H., Weiner, A.J.: The Year 2000: A Framework for Speculations on the Next Thirty-Three Years. The MacMillan Company, 1967

[Kirschner et al., 2003] Kirschner, P.A., Buckingham Shum, S.J. and Carr, C.S.: Visualizing Argumentation: Software Tools for Collaborative and Educational Sense-Making, Springer-Verlag, London, UK., 2003

[Kolabora, 2007] Kolabora: "Collaborative Writing Tools And Technology: A Mini-Guide". http://www.kolabora.com/news/2007/03/01/collaborative_writing_tools_and_technology.htm [last access on 10.09.2010]

[Koenig, 2010] König, T.: CAQDAS Comparison. 2010, http://www.restore.ac.uk/lboro/research/software/ caqdas_comparison.php#fl1995 [last access on 13.09.2010]

[Kuckartz, 2007] Kuckartz, U.: Einführung in die computergestützte Analyse qualitativer Daten. Vs Verlag für Sozialwissenschaften, 2nd edition., revised and advanced edition. (16 January 2007), ISBN-13: 978-3531342474

[May, 1996] May, G.: The future is ours. London: Adamantine Press, 1996

[Mayring, 2007] Mayring, P.: Qualitative Inhaltsanalyse – Grundlagen und Techniken. 9. Edition. Beltz Verlag, 2007

[Nam, 2004] Nam, M.: Systematic scenario walkthroughs with ARTSCENE. In Scenarios, stories, use cases: through the systems development life-cycle, I. Alexander , M. Nam Eds. John Wiley, New York, 2004, 161–178.

[OECD, 2008] OECD Environmental Outlook to 2030, March 2008, ISBN: 9789264040489. http://www.oecd.org/document/20/0,3343,en_2649_34305_39676628_1_1_1_37465,00.html

[Prakken and Vreeswijk, 2002] Prakken, H., Vreeswijk, G.A.W.: Encoding schemes for a discourse support system for legal argument. In: Workshop Notes of the ECAI-02 Workshop on Computational Models of Natural Argument, 2002, 31–39.

[Reed and Rowe, 2004] Reed, C.A. and Rowe, G.A.W.: Araucaria: Software for argument analysis, diagramming and representation. In: International Journal on Artificial Intelligence Tools, 2004, 14(3-4), 961–980

[Schank and Ranney, 1995] Schank, P. and Ranney, M.: Improved reasoning with Convince Me. In: CHI ’95: Conference Companion on Human Factors in computing Systems, 276–277, New York, NY. ACM Press, 1995

[Selvin et al., 2001] Selvin, A., Buckingham Shum, S., Sierhuis, M., Conklin, J., Zimmermann, B., Palus, C., Drath, W., Horth, D., Domingue, J., Motta, E., Li, G.: Compendium: Making meetings into knowledge events. In: Proceedings Knowledge Technologies 2001.

[Sharpe and Van der Heijden, 2007] Sharpe, B., Van der Heijden, K.: Scenarios for Success – Turning Insights into Action. John Wiley &Sons, Ltd., England, 2007.

[Straeter, 1988] Sträter, D.: Szenarien als Instrument der Vorausschau in der räumlichen Planung. In Akademie für Raumforschung und Landesplanung (Ed.): Regionalprognosen. Methoden und ihre Anwendung, 1988, 417-440, Hannover (Veröffentlichungen der Akademie für Raumforschung und Landesplanung: Forschungs- und Sitzungsberichte, 175).

[Suthers et al., 1995] Suthers, D., Weiner, A., Connelly, J., Paolucci, M.: Belvedere: Engaging students in critical discussion of science and public policy issues. In: AI-Ed 95, the 7th World Conference on Artificial Intelligence in Education, 1995, 266–273

[Tietje, 2008] Tietje, O.: SCrategy Projektentwicklung, 2008, http://www.scrategy.com/Software/index.htm

[Surrey, 2010] University of Surrey: QUIC Briefing and Working Papers, 2010, http://caqdas.soc.surrey.ac.uk/Resources/QUICworkingpapers.html [last access on 13.09.2010]

[Van den Herik and de Vreede, 2000] van den Herik, C.W.; de Vreede, G.J.: Experiences with Facilitating Policy Meeting with Group Support Systems. In International Journal of Technology and Management, 19, 2000, 2/3/4, 246-268.

[Van der Duin et al., 2001] Van der Duin, P.A., Drop, R. and Kloosterhof, A.: The world of future studies according to KPN Research. Leidschendam: KPN Research, 2001.

[Van Gelder, 2002] van Gelder, T.J.: Argument mapping with Reason!Able. The American Philosophical Association Newsletter on Philosophy and Computers, 2002, 85–90

[Van Notten et al., 2003] van Notten, P.W.F., Rotmans, J., van Asselt, M.B.A. and Rothmann, D.: An updated scenario typology. In: Futures, 35, 2003, 423-443

[Verheij, 2003] Verheij, B.: Artificial argument assistants for defeasible argumentation. Artificial Intelligence, 2003, 150(1-2), 291–324

[Von Reibnitz, 1987] Von Reibnitz, U.: Szenarien - Optionen für die Zukunft. Hamburg, McGraw-Hill, 1987

[Weitzman and Miles, 1995] Weitzman, E., Miles M.B.: Computer Programs for Qualitative Analysis. A Software Sourcebook. Sage. London, 1995.

Formal modelling tools and technologies

[Berryman, 2008] Berryman, M.: Review of Software Platforms for Agent Based Models. Technical report DSTO-GD-0532, Department of Defence, Defence Science and Technology Organisation, Land Operations Division, Edinburgh, Australia, April 2008.

[Brazier et al., 1997] Brazier, F.M.T., Dunin-Keplicz, B.M., Jennings, N.R. and Treur, J.: DESIRE: Modelling Multi-Agent Systems in Compositional Formal Framework. Int. Journal of Cooperative Information Systems, 1997, vol 6, 67-94.

[Caves, 2006] CAVES Periodic Activity Report, Project Month 18; FP7 Project 012816: CAVES – Complexity, Agents, Volatility, Evidence and Scale, 2006, http://cfpm.org/caves

[Epstein and Axtell, 1996] Epstein, J. and Axtell, R.: Growing Artificial Societies: Social Science from the Bottom-Up. MIT Press, 1996.

[Figueira and Ramalho, 2000] da Figueira Filho, S.C. and Ramalho, L. G.: JEOPS — The Java Embedded Object Production System. In: Advances in Artificial Intelligence, Springer, 2000, 53-62 (Lecture Notes in Computer Science; 1952)

[Forgy, 1982] Forgy, C.: Rete: a Fast Algorithm for the Many Pattern/Many Objects Pattern Match Problem. In Artificial Intelligence, v. 19, 1982, 17-37

[Gilbert and Bankes, 2002] Gilbert, N. and Bankes, S.: Platforms and methods for agent-based modelling. PNAS 99 (supp. 3):7197-7198, 2002.

[Gilbert and Troitzsch, 2005] Gilbert, N. and Troitzsch, K.G.: Simulation for the Social Scientist. 2nd edition. Open University Press/Mc Graw Hill, 2005.

[jsr-94, 2004] JSR-94: Java Specification Request #94: Java Rule Engine API. Final release, August 2004. http://www.jcp.org/en/jsr/detail?id=94

[Luke et al., 2005] Luke, S., Cioffi-Revilla, C., Panait, L., Sullivan, K. and Balan, G.: Mason: A multiagent simulation environment. Simulation 2005, 81(7), 517-527. http://dx.doi.org/10.1177/0037549705058073.

[Minar et al., 1996] Minar, N., Burkhart, R., Langton, C. and Askenazi, M.: The Swarm simulation system: a toolkit for building multi-agent simulations. Technical report, 1996.

[Moss and Edmonds, 2005] Moss, S. and Edmonds, E.: Sociology and simulation: statistical and qualitative cross-validation. American Journal of Sociology, 2005, 11(4):1095-1131.

[Moss et al.,1998] Moss, S., Gaylard, H, Wallis, S. and Edmonds, B.: SDML: A Multi-agent Language for Organizational Modelling. Computational and Mathematical Organization Theory, 1998, 4 (1), 43-69.

[Najlis et al.,2001] Najlis, R., Janssen, M. A. and Parker, D. C.: Software tools and communication issues. Proc. Of the Agent-Based Models of Land-Use and Land-Cover Change Workshop, 2001, 17-30.

[Nikolai and Madey, 2009] Nikolai, C. and Madey, G.: Tools of the Trade: A Survey of Various Agent Based Modelling Platforms. Journal of Artificial Societies and Social Simulation, 2009, 12(2), http://jasss.soc.surrey.ac.uk/12/2/2.html

[North et al., 2006] North, M. J., Collier, N. T. and Vos, J. R.: Experiences creating three implementations of the repast agent modelling toolkit. ACM Trans. Model. Comput. Simul., 2006, 16(1), 1-25.

[Parker, 2001] Parker M.: What is Ascape and Why Should You Care? Journal of Artificial Societies and Social Simulation, 2001, 4 (1) 5, http://jasss.soc.surrey.ac.uk/4/1/5.html

[Railsback et al., 2006] Railsback, S.F., Lytinen, S.L., and Jackson, S.K.: Agent-based simulation platforms: Review and development recommendations. Simulation, 2006, 82(9):609-623, http://www.humboldt.edu/ecomodel/documents/ABMPlatformReview.pdf

[Tobias and Hofmann, 2004] Tobias, R . and Hofmann, C.: Evaluation of free Java-libraries for social-scientific agent based simulation. Journal of Artificial Societies and Social Simulation, 2004, 7 (1) 6, http://jasss.soc.surrey.ac.uk/7/1/6.html

[Wilensky, 1999] Wilensky, U.: NetLogo. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL, 1999, http://ccl.northwestern.edu/netlogo/

Architecture

[Booch et al., 2007] Grady Booch, Robert A. Maksimchuk, Michael W. Engle, Bobbi J. Young, Jim Conallen, Kelli A. Houston: Object-Oriented Analysis and Design with Applications, Third Edition. Addison Wesley Professional, 2007.

[Coad and Yourdon, 1991] Peter Coad, Edward Yourdon: Object-oriented design. Yourdon Press, Upper Saddle River, NJ, USA, 1991, ISBN 0-13-630070-7.

[Esteves, 2007] Esteves, J.: A Semiotic Analysis of Spanish Local e-Government Websites. In: Proceedings of the 7th European Conference on E-Government 200, ECEG, Academic Conferences Ltd., 2007.

[Firesmith, 2002] Firesmith, D.G.: The OPEN process framework, Pearson Educatrion Limitd, 2002, ISBN 0-201-67510-2.

[IEEE, 2000] IEEE Computer Society. Recommended Practice for Architectural Description of Software-Intensive Systems. IEEE Std-1471-2000, October 9, 2000.

[Mayhew, 1999] Mayhew, D.J.: The Usability Engineering Lifecycle: A practitioner's handbook for user interface design. San Francisco, Calif., Morgan Kaufmann, 1999.

[Nielsen, 1993] Nielsen, J.: Usability Engineering. Boston, Mass., Acad. Press, 1993.

[Rozanski and Woods, 2005] Nick Rozanski, Eoin Woods: Software Systems Architecture. Working with Stakeholders Using Viewpoints and Perspectives. Addison Wesley, 2005.

[Scherer et al., 2009] Scherer, S., Karamagioli, E., Titorencu, M., Schepers, J., Wimmer, M.A and Koulolias, V.: Usability Engineering in eParticipation. In: European Journal of ePractice, Nr. 7, 2009, 79-91.

Relevant standards

[AMQP] AMQP 1.0 revision 0, Recommendation. AMQP Working Group, 17. August 2010. http://www.amqp.org

[ARIA_W3C] Accessible Rich Internet Applications (WAI-ARIA) 1.0. W3C Working Draft, 16 September 2010. http://www.w3.org/TR/wai-aria/

[BPEL4P_OASIS] WS-BPEL Extension for People (BPEL4People) Specification, Version 1.1. Committee Specification, 17 August 2010. http://docs.oasis-open.org/bpel4people/bpel4people-1.1.html

[BPMN1.2_OMG] Business Process Model and Notation (BPMN), Version 1.2. OMG Standard, January 2009. http://www.omg.org/spec/BPMN/1.2/

[BPMN2.0_OMG] Business Process Model and Notation (BPMN), Version 2.0 Beta 2. OMG Working Draft, May 2010. http://www.omg.org/spec/BPMN/2.0/

[CMIS_OASIS] Content Management Interoperability Services (CMIS), Version 1.0. OASIS Standard, 1 May 2010. http://docs.oasis-open.org/cmis/CMIS/v1.0/cmis-spec-v1.0.html

[ConnA_JSR112] JSR 112: J2EE Connector Architecture 1.5. Final Release, 24 December 2003. http://www.jcp.org/en/jsr/detail?id=112

[CORBA3.1_OMG] - Common Object Request Broker Architecture (CORBA), Version 3.1. OMG Standard, January 2008. http://www.omg.org/spec/CORBA/3.1/

[EJB_JSR220] JSR 220: Enterprise JavaBeans 3.0. Final Release, 11 May, 2006. http://www.jcp.org/en/jsr/detail?id=220

[JBI_JSR208] JSR 208: Java Business Integration (JBI). Final Release, 25 August 2005. http://jcp.org/en/jsr/summary?id=208

[JDBC_JSR54] JSR 54: JDBC 3.0 Specification. Final Release, 09 May 2002. http://www.jcp.org/en/jsr/detail?id=54

[JDO_JSR12] JSR 12: Java Data Objects (JDO) Specification. Final Release 2, 16 September 2003. http://www.jcp.org/en/jsr/detail?id=12

[JMS] Java Message Service, Version 1.1. 12 April 2002. http://www.oracle.com/technetwork/java/index-jsp-142945.html

[JPA_JSR317] JSR 317: Java Persistence 2.0. Final Release, 10 December 2009. http://www.jcp.org/en/jsr/detail?id=317

[JRE_JSR94] JSR 94: Java Rule Engine API. Final Release, 04 August 2004. http://jcp.org/en/jsr/summary?id=94

[OWLS_W3C] OWL-S: Semantic Markup for Web Services. W3C Member Submission, 22 November 2004. http://www.w3.org/Submission/OWL-S/

[Port_JSR286] JSR 286: Portlet Specification 2.0. Final Release, 12 Jun 2008. http://www.jcp.org/en/jsr/detail?id=286

[RDFS_W3C] RDF Vocabulary Description Language 1.0: RDF Schema. W3C Recommendation, 10 February 2004. http://www.w3.org/TR/rdf-schema/

[RFSOA_OASIS] Reference Model for Service Oriented Architecture, Version 1.0. OASIS Standard, 12 October 2006. http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.html

[UDDI_OASIS] UDDI Version 3.0.2. UDDI Spec Technical Committee Draft, 19 October 2004. http://uddi.org/pubs/uddi_v3.htm

[UML2.3_OMG] Unified Modelling Language (UML), Version 2.3. OMG Standard, May 2010. http://www.omg.org/spec/UML/2.3

[SAML_OASIS] Security Assertion Markup Language (SAML), Version 2.0. OASIS Standard, 15 March 2005. http://www.oasis-open.org/specs/#samlv2.0

[SAWSDL_W3C] Semantic Annotations for WSDL and XML Schema. W3C Recommendation, 28 August 2007. http://www.w3.org/TR/sawsdl/

[SConn_JSR279] JSR 279: Service Connection API for Java ME. Final Release, 16 November 2009. http://www.jcp.org/en/jsr/detail?id=279

[SOAP_W3C] SOAP Version 1.2 Part 1: Messaging Framework (Second Edition). W3C Recommendation, 27 April 2007. http://www.w3.org/TR/soap12/

[SysML_OMG] OMG Systems Modelling Language (SysML), Version 1.2. OMG Standard, June 2010. http://www.omg.org/spec/SysML/1.2/

[WCAG_W3C] Web Content Accessibility Guidelines (WCAG) 2.0. W3C Recommendation, 11 December 2008. http://www.w3.org/TR/WCAG20/

[WebDAV] RFC 4918: HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV). The IETF Trust, 2007. http://www.webdav.org/specs/rfc4918.html

[WSAddr_W3C] Web Services Addressing 1.0 - Core. W3C Recommendation, 9 May 2006. http://www.w3.org/TR/ws-addr-core/

[WSBPEL_OASIS] Web Services Business Process Execution Language, Version 2.0. OASIS Standard, 11 April 2007. http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html

[WSDL_W3C] Web Services Description Language (WSDL), Version 2.0. Part 1: Core Language. W3C Recommendation, 26 June 2007. http://www.w3.org/TR/wsdl20/

[WSEn_W3C] Web Services Enumeration. W3C Working Draft, 5 August 2010. http://www.w3.org/TR/ws-enumeration/

[WSEv_W3C] Web Services Eventing. W3C Working Draft, 5 August 2010. http://www.w3.org/TR/ws-eventing/

[WSEvDesc_W3C] Web Services Event Descriptions. W3C Working Draft, 5 August 2010. http://www.w3.org/TR/ws-event-descriptions/

[WSFrag_W3C] Web Services Fragment (WS-Fragment). W3C Working Draft, 5 August 2010. http://www.w3.org/TR/ws-fragment/

[WSMdE_W3C] Web Services Metadata Exchange. W3C Working Draft, 5 August 2010. http://www.w3.org/TR/ws-metadata-exchange/

[WSML_W3C] Web Service Modelling Language (WSML). W3C Member Submission, 3 June 2005. http://www.w3.org/Submission/WSML/

[WSPolicy_W3C] Web Services Policy 1.5 - Framework. W3C Recommendation, 04 September 2007. http://www.w3.org/TR/ws-policy/

[WSResTr_W3C] Web Services Transfer. W3C Working Draft, 5 August 2010. http://www.w3.org/TR/ws-transfer/

[WSRP_OASIS] Web Services for Remote Portlets Specification, Version 2.0. OASIS Standard, 1 April 2008. http://docs.oasis-open.org/wsrp/v2/wsrp-2.0-spec.html

[XACML_OASIS] eXtensible Access Control Markup Language (XACML), Version 2.0. OASIS Standard, 1 February 2005. http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf

[XForms_W3C] XForms 1.1. W3C Recommendation, 20 October 2009. http://www.w3.org/TR/xforms/

[XPath_W3C] XML Path Language (XPath), Version 1.0. W3C Recommendation, 16 November 1999. http://www.w3.org/TR/xpath/

[XQuery_W3C] XQuery 1.0: An XML Query Language. W3C Recommendation, 23 January 2007. http://www.w3.org/TR/xquery/

[XSLT_W3C] XSL Transformations (XSLT), Version 1.0. W3C Recommendation, 16 November 1999. http://www.w3.org/TR/xslt

APPENDIX A: RELEVANT STANDARDS

The compliance with widely accepted standards is a natural requirement for development of almost all type of product, including software systems such as OCOPOMO. It is, however, important to note that the correspondence with standards is not obligatory; an application of particular standards is in most cases voluntary and optional. Nevertheless, acceptance of standards during the design and development is highly recommended, since it can bring a competition advantage for the resulting product. For software systems it can, in addition, ensure a compatibility and interoperability with other standardised solutions.

This section provides an outline of standards that are relevant to the OCOPOMO platform on both conceptual and technological levels. It summarises the standardised frameworks and approaches of integration technology background, which were mentioned or referenced in chapter 3.1. In addition, standards for modelling technologies, Web services, IT service management, and other related fields are presented and briefly described. The list of standards is organised according to particular vendors, standardisation bodies such as ISO, IEC, W3C, OMG, OASIS, including technology providers such as Sun/Oracle that provide some of relevant industrial standards.

A.1. ISO AND ISO/IEC STANDARDS

The International Organization for Standardization (ISO)[134] is probably the most authoritative international standardisation body, which proposes and publishes world-wide proprietary industrial and commercial standards. It covers practically all the areas where the regulation by standards can be applied. Namely, assuming the fields of interest of OCOPOMO, it includes such areas as information technology, quality management, energy systems and renewable energy sources, financial services, environmental management, and many others.

The standardisation work of ISO is organised into technical committees (TC), subcommittees (SC) and working groups (WG), which involve participants of national standardisation organisations as well as of other relevant standardisation bodies of particular areas. For example, the broad area of information technologies is processed by the Joint Technical Committee No. 1 (JTC 1) of ISO and International Electrotechnical Commission (IEC[135]). The world-wide acceptance of ISO standards is ensured by close co-operation with national standardisation bodies of ISO member countries and with numerous organisations including, for example, European Commission, ANSI, United Nations, UNESCO, etc.

The OCOPOMO platform in the aspects of architectural design, software engineering, data storage and access, documentation development, system operation and maintenance may be supported by ISO standards of information technology, which are produced by the ISO/IEC JTC 1 joint committee and its subcommittees. Some of the most relevant standards published in this field are listed in the following outline:

JTC 1 : Information technology

- ISO/IEC 29361-29363:2008 *Web Services Interoperability*. Three standards define the Web services profiles, consisting of a set of non-proprietary specifications, along with clarifications and amendments to those specifications that are intended to promote interoperability.

[134] http://www.iso.org

[135] http://www.iec.ch

- ISO/IEC TR 29138:2009 *Accessibility considerations for people with disabilities*. Three technical reports (TR), which identify a collection of user needs of people with disabilities for documentation developers to take into consideration when developing or revising the documentation materials, together with a guidance on meeting these needs.

SC 7 : Software and systems engineering

- ISO/IEC TR 9126:2001-04 *Product quality*. The set of standards provides six external quality characteristics and internal metrics for attribute-based measuring of the overall quality of software systems.
- ISO/IEC 20000:2005-10 *Service management*. The set of standards, which is based on British standard BS 15000, defines the requirements for a service provider to deliver managed IT services. By adopting structured, proactive working practices, it enables service providers to understand how to enhance the quality of service delivered to their customers, both internal and external.

SC 32 : Data management and interchange

- ISO/IEC 9075:2008 *Database languages - SQL*. The set of standards defines the SQL language, including the conceptual framework, formal grammar, data structure and the operations on data stored in that structure.
- ISO/IEC TR 10032:2003 *Reference Model of Data Management*. The standard defines common terminology and concepts related to all data held within information systems.
- ISO/IEC 15944:2002-09 *Business Operational View*. The set of standards addresses the identification, registration, referencing and reusability of common objects, scenarios and scenario components, in a business transaction.
- ISO/IEC 19763:2010 *Metamodel framework for interoperability*. It specifies a metamodel that provides a facility to register administrative and evolution information related to ontologies, independent of the languages in which they are expressed. The metamodel also administers the authoritative extent of ontologies, which indicates how commonly they can be used.

SC 34 : Document description and processing languages

- ISO/IEC 19757:2006-09 *Document Schema Definition Language*. A definition of a set of languages that can be used to specify one or more validation processes performed against XML or SGML documents.
- ISO/IEC 13250:2006-09 *Topic Maps*. The set of standards specifies the Topic Maps data model. It defines an abstract structure and interpretation of topic maps, the rules for merging topic maps and a set of fundamental subject identifiers.

When applying the IT infrastructure in an environment of a particular organisation (e.g. setting up the OCOPOMO system in a municipality), the quality of the whole solution needs to be assured on a standardised level. The quality management of IT services is specified by the above-mentioned ISO/IEC 20000 standards. In addition, ISO provides a set of standards on general quality management in an organisation, which are generated and maintained within the TC 176 Quality management and quality assurance. The documents published in this TC define basic concepts and terminology of quality management systems, specify quality requirements, supported technologies, and guidelines for applying these systems in practice. These issues are published in well-known and widely accepted standards of ISO 9000 family, including, for example:

- ISO 9000:2005 Fundamentals and vocabulary of quality management systems.
- ISO 9001:2008 Requirements on quality management systems.
- ISO 9004:2009 Managing for the sustained success of an organization - A quality management approach.
- ISO 10005-10007:2003-05 Guidelines for quality plans, quality management in projects, and configuration management.
- ISO/TR 10013:2001 Guidelines for quality management system documentation.
- ISO 10014:2006 Guidelines for realizing financial and economic benefits.
- ISO 19011:2002 Guidelines for quality and/or environmental management systems auditing, etc.

The ISO structure also contains several standardisation areas that should be taken into consideration by the OCOPOMO pilot applications. It namely includes TC 68 financial services, TC 154 processes data elements and documents in administration (Campania pilot), TC 207 environmental management, TC 180 solar energy, TC 203 technical energy systems, and ISO/IEC NP 13273 energy efficiency and renewable energy sources (KSR pilot). However, these standards are not directly related to the system architecture and thus a more detailed description will be omitted here.

A.2. W3C STANDARDS AND RECOMMENDATIONS

The World Wide Web Consortium (W3C)[136] is an international community that creates, publishes and maintains standardisation documents for the World Wide Web. It is nowadays the main world-wide authority in this field, with its 329 member organisations (September 2010). The W3C standards include the areas of Web design and applications, Web architecture, Semantic web, XML technology, Web of services and SOA, Web of devices, Web browsers and authoring tools. Specifications related to web services, XML schemas or web accessibility, which particularly may be considered during the development of OCOPOMO architecture, are presented in the following outline.

Web services and service-related issues, which are provided by the W3C Web Services Activity[137]:

- SOAP (Simple Object Access Protocol) is specified in the W3C Recommendation *SOAP Version 1.2* [SOAP_W3C], which was issued in April 27, 2007. It introduces the lightweight protocol intended for exchanging structured information in a decentralised, distributed environment. The standard consists of four documents that provide messaging framework (Part 1), adjuncts as data model, encoding and RPC representation (Part 2), assertions and test collection (Part 3), and the SOAP processing model with basic usage scenarios (Part 0).
- WSDL (Web Services Description Language) is specified in the W3C Recommendation *Web Services Description Language Version 2.0* [WSDL_W3C], issued in June 26, 2007. The standard provides a model and an XML format for describing Web services. It defines a language for describing the abstract functionality of a service as well as a framework for describing the concrete details of a service description. The companion specification of WSDL Part 2: Adjuncts describes extensions for message exchange patterns, operation safety, operation styles and binding extensions.

[136] http://www.w3.org

[137] http://www.w3.org/2002/ws/Activity

- Additional specifications for web services include recommendations and working drafts on addressing [WSAddr_W3C], policy [WSPolicy_W3C], resource transfer [WSResTr_W3C], metadata exchange [WSMdE_W3C], eventing [WSEv_W3C], enumeration [WSEn_W3C], fragmenting [WSFrag_W3C], and event descriptions [WSEvDesc_W3C].

Various XML-based languages and schemas for transforming and querying the data sources in XML format, including the WSDL representation of web services, for example:

- XPath, specification of a formal language for addressing parts of an XML document [XPath_W3C];
- XSLT, definition of the syntax and semantics of a language for transforming XML documents into other XML documents [XSLT_W3C];
- XQuery, specification of a query language over XML documents and data sources of various types [XQuery_W3C].
- XForms, a new platform-independent mark-up language for on-line interaction between a person and another, usually remote, agent [XForms_W3C]. XForms are the successor to HTML forms and are aiming at providing dynamism, multi-modality, and device independence of web applications.

Semantic extensions of web services and the concept of Semantic web in general are supported by:

- SAWSDL recommendation [SAWSDL_W3C] defines a set of extension attributes for WSDL and XML Schema languages that allows description of additional semantics of web services. The specification defines how semantic annotation is accomplished using references to semantic models, e.g. ontologies. It provides mechanisms by which concepts from the semantic models, typically defined outside the WSDL document, can be referenced from within WSDL and XML Schema components using annotations.
- Several ontology formats such as RDF Schema [RDFS_W3C], OWL-S (Semantic Markup for Web Services) [OWLS_W3C], or WSML (Web Service Modelling Language) [WSML_W3C].

W3C Web Accessibility Initiative (WAI)[138] is aiming at developing strategies, guidelines, and resources to help make the web accessible to people with disabilities. With the respect of designing and building applications with web-based interfaces, as it is assumed for the OCOPOMO platform, the most relevant is the WAI standardisation of web content:

- *Web Content Accessibility Guidelines* [WCAG_W3C] cover a wide range of recommendations for making web content more accessible to people with disabilities, including blindness and reduced vision, deafness and hearing loss, learning disabilities, cognitive limitations, limited movement, speech disabilities, photosensitivity and combinations of these.
- *Accessible Rich Internet Applications* working draft [ARIA_W3C] provides an ontology of roles, states, and properties that define accessible user interface elements and can be used to improve the accessibility and interoperability of web content and applications.

[138] http://www.w3.org/WAI/

A.3. OASIS STANDARDS

The Organization for the Advancement of Structured Information Standards (OASIS)[139] is a not-for-profit consortium that drives the development, convergence and adoption of open standards for the global information society. It produces fundamental and advanced Web services standards along with standards for security, e-business, and standardisation efforts in the public sector and for application-specific markets. Nowadays, OASIS has more than 5000 participants representing over 600 organisations and individual members in 100 countries.

The following list contains a selection of OASIS standards that might be of particular concern and relevance of OCOPOMO, covering various aspects of business processes, interoperability, content management, security policy issues and architecture of Web services applications.

- *Universal Description, Discovery and Integration* (UDDI) is an XML-based registry for businesses world-wide to list themselves on the Internet. The UDDI Version 3.0.2 specification [UDDI_OASIS] describes the Web services, data structures and behaviours of all instances of a UDDI registry.
- The *Content Management Interoperability Services* (CMIS) specification [CMIS_OASIS] defines a domain model and Web services bindings that can be used by applications to work with one or more Content Management repositories or systems.
- The *Web Services Business Process Execution Language* (WS-BPEL) specification [WSBPEL_OASIS] defines a language for specifying business process behaviour based on Web services. Processes in WS-BPEL export and import functionality by using Web service interfaces exclusively. The WS-BPEL language is meant to be used to model the behaviour of both executable and abstract processes.
- The *WS-BPEL Extension for People* (BPEL4People) specification [BPEL4P_OASIS] introduces a process modelling extension to address human interactions in WS-BPEL. It defines a new type of basic activity which uses human tasks as an implementation, and allows specifying tasks local to a process or using tasks defined outside of the process definition.
- Information security policy and access control is supported by the *XACML mark-up language* [XACML_OASIS] which should allow managing and enforcing the elements of a security policy in all components of information systems in an enterprise or institution. Managing security policy may include steps such as writing, reviewing, testing, approving, issuing, combining, analysing, modifying, withdrawing, retrieving and enforcing policy.
- Security-related issues such as single sign-on, user authentication, entitlement, and attribute information are supported by the Security Assertion Markup Language (SAML). This XML-based framework, currently available in version 2.0 [SAML_OASIS], allows creating and exchanging security information between on-line partners in web-based and/or service-oriented applications.
- The *Web Services for Remote Portlets* (WSRP) specification [WSRP_OASIS] defines a set of interfaces and related semantics which standardise interactions with components providing user-facing mark-up, including the processing of user interactions with that mark-up. This allows applications to consume such components as providing a portion of the overall user application without having to write unique code for interacting with each component.
- The *Reference Model for Service Oriented Architecture* standard [RFSOA_OASIS] provides an abstract framework for understanding significant entities and relationships between them

[139] http://www.oasis-open.org

within a service-oriented environment, and for the development of consistent standards or specifications supporting that environment. It is based on unifying concepts of SOA and may be used by architects developing specific service oriented architectures or in training and explaining SOA.

In addition, OASIS provides a whole suite of standards proposing XML-based schemas and frameworks for Web services applied in specific domains as, for example, healthcare, electoral processes, general industry data formats, emergency information systems, library of books and papers about computer hardware and software, etc.

A.4. OMG STANDARDS

The Object Management Group (OMG)[140] is an international, open membership, not-for-profit computer industry consortium, which was originally aimed at setting standards for distributed object-oriented systems. Nowadays, OMG activities are mostly focused on modelling (programs, systems and business processes) and provisioning of model-based standards – some of them, which may be relevant for the OCOPOMO architecture design, are listed below.

OMG provides several well-known specifications for object request brokers and frameworks for data interchange, which include:

- *Common Object Request Broker Architecture* (CORBA), the specification of an architecture for middleware technology that provides interoperability among clients and servers distributed over a heterogeneous environment [CORBA3.1_OMG]. CORBA includes a set of additional specifications for component model, reflective operations for objects, interfaces, protocols, etc.
- A set of OMG metadata specifications such as *XML Metadata Interchange*, *Common Warehouse Metamodel*, *Meta Object Facility*, *Ontology Definition Metamodel* and several other models[141].

Modelling of business processes and complex systems is supported by OMG notations, frameworks and techniques such as:

- The *Business Process Model and Notation* (BPMN)[142] is a specification of a graphical format for modelling abstract business processes. The version 1.2 of BPMN was released as the OMG standard [BPMN1.2_OMG], while the next version 2.0 is still under development and is available as a working draft [BPMN2.0_OMG].
- *The Unified Modelling Language* (UML)[143] is a popular and frequently used set of graphical and formal notations for general-purpose modelling. It is particularly applicable in the field of software engineering, but can be employed for modelling of business processes and service-based systems of various types. Currently it exists in the version 2.3 [UML2.3_OMG], where the OMG standard includes two complementary specifications: Infrastructure (defines the

[140] http://www.omg.org

[141] http://www.omg.org/technology/documents/modeling_spec_catalog.htm

[142] http://www.bpmn.org

[143] http://www.uml.org

foundational language constructs) and Superstructure (defines the user level constructs). The UML of version 1.4.2 was released as the ISO/IEC 19501 standard.

- The *OMG Systems Modelling Language* is a general-purpose graphical modelling language for specifying, analyzing, designing, and verifying complex systems [SysML_OMG]. It provides graphical representations with a semantic foundation for modelling system requirements, behaviour, structure, and parameters, which is used to integrate with other engineering analysis models. It is a subset of UML ver. 2 with extensions needed to satisfy the requirements of the UML applied for systems engineering.

The above-listed OMG specifications are conceptually integrated into a proposal of the *Model Driven Architecture* (MDA)[144], which is a set of specifications that provide an open, vendor-neutral approach for developing enterprise information systems. MDA separates business and application logic from underlying platform technology. It is based on UML models and other related OMG modelling standards, then it can be realised through any open or proprietary platform including Web services, CORBA, .NET, J2EE, etc.

A.5. OTHER OPEN OR INDUSTRIAL STANDARDS

The OCOPOMO platform is proposed to be built upon the Java-related technologies, which were originally produced by Sun Microsystems and currently are provided by Oracle. This proposal implies the relevance of Java standards that are provided as Java Specification Requests (JSRs)[145]. Below we present a selection of the industrial standards issued and maintained by Sun/Oracle and few other vendors, which are organised according to the layers of data, business logic, and user interface.

Database storage, access, and connectivity:

- The *Java Database Connectivity* (JDBC) API is the industry standard for database-independent connectivity between the Java programming language and a wide range of relational SQL databases and other tabular data sources. Latest version of JDBC is published in the JSR 54: JDBC 3.0 Specification [JDBC_JSR54].
- The *Java Data Objects* (JDO) specification, published as JSR 12: JDO Specification [JDO_JSR12], provides definitions of data stores and transactions, together with a description of selection and transformation of persistent storage data into native Java objects.
- The *Java Persistence API*, published in JSR 317: Java Persistence 2.0 [JPA_JSR317], is the Java API for the management of persistence and object/relational mapping for Java enterprise and standard environments.

Message-oriented middleware, business logic and data processing:

- The *Java Message Service* (JMS) API is a messaging standard [JMS] that allows application components based on the Java 2 Platform, Enterprise Edition (J2EE) to create, send, receive, and read messages. It enables distributed communication that is loosely coupled, reliable, and asynchronous. With release 1.4 of the J2EE platform, the JMS provider may be integrated with the application server using the J2EE Connector Architecture [ConnA_JSR112].

[144] http://www.omg.org/mda/

[145] http://www.jcp.org/en/jsr/

- The *Java Service Connection API*, published as JSR 279 [SConn_JSR279], is a new high-level API for connection services via frameworks supporting identity-based services, discovery, and authentication. The API supports Service Oriented Architectures and other similar network service application models.
- The *Java Business Integration*, provided as JSR 208 [JBI_JSR208], is a specification of handling the principles of service oriented architecture and enterprise service bus in Java implementations. Version 2.0, referenced as JSR 312, is currently under development.
- The *Advanced Message Queuing Protocol* (AMQP) is an emerging open standard [AMQP] that defines the protocol and formats used for business messaging and information exchange between server and client. It is provided and maintained by the AMQP Working Group[146].
- The *Java Rule Engine API*, referenced as JSR 94 [JRE_JSR94], defines a lightweight-programming interface that constitutes a standard API for acquiring and using a rule engine.

Architectures, content and presentation integration:

- The *Enterprise JavaBeans* specification, published as JSR 220: Enterprise JavaBeans 3.0 [EJB_JSR220], defines an architecture for the development and deployment of component-based business applications. It is a server-side model that encapsulates the business logic of an application and supports the features such as scalability, transactional data access and multi-user security.
- The *Portlet Specification*, provided as JSR 286: Portlet Specification 2.0 [Port_JSR286], defines an API for portlets - web-based components that enable integration between applications and web portals. It also provides a portlet driver, which is a lightweight portlet rendering environment.
- The *Web-based Distributed Authoring and Versioning* was prepared and published by the IETF Trust as the RFC 4918 specification [WebDAV]. It defines a set of HTTP-based methods, headers, and content-types for the management of web resource properties, creation and management of resource collections, URL namespace manipulation, and resource locking. It is accompanied with a set of specifications that enhance its basic functionality with extensions of search, versioning, binding, calendaring, access control, etc.

[146] http://www.amqp.org

APPENDIX B: CMS COMPARISON

The following overview shows a comparison between Alfresco, Drupal, Joomla, Typo3, Plone, XOOPs, and Wordpress as retrieved from http://www.cmsmatch.com [accessed on 16th September, 2010][147]. Please, note that information is not provided for the latest version in particular for Alfresco. Moreover, the comparison is made in a general (not OCOPOMO related) manner considering also features not important for the project.

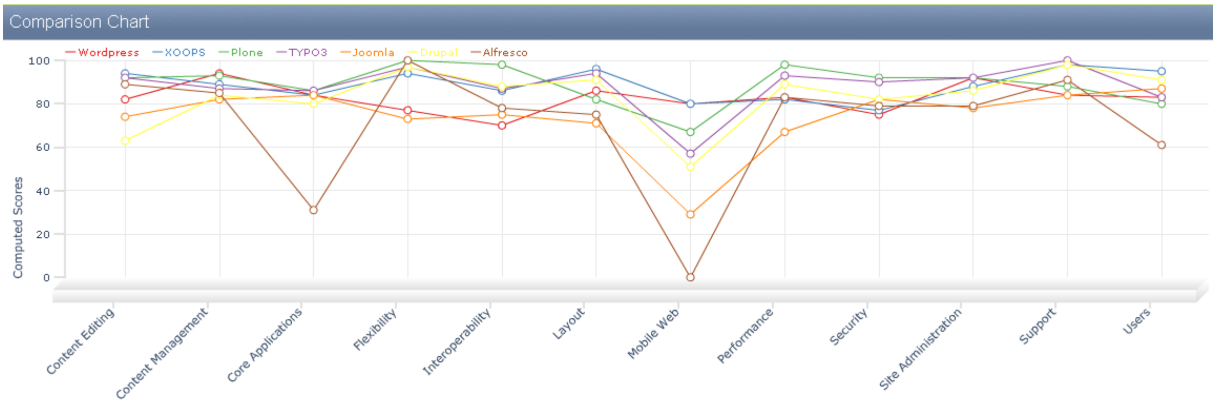


Figure 62 Summary of CMS comparison

Legend:

- No: feature not available; *0 points*
- Limited (Paid): basic feature at extra cost; *1 point*
- Limited (Add-on): basic feature, but as free plugin; *3 points*
- Limited: free and included, but still basic; *5 points*
- Yes (Paid): advanced but at extra cost; *6 points*
- Yes (Add-on): advanced free plugin; *8 points*
- Yes: the full works; *10 points*

	Wordpress	**XOOPS**	**Plone**	**TYPO3**	**Joomla**	**Drupal**	**Alfresco**
Software Details							
Stability	Mature	Mature	Mature	Mature	Mature	Mature	
Version	2.9.x	2.x	3.x	4.3beta1	1.5.x	6.15	0.x
Meta Score	83	89	89	88	74	83	77
Completed Listing	92%	98%	97%	98%	97%	100%	71%
Cost Estimate	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00	$15000.00
Available Languages	65	11	62	49	81	51	
Learning Curve	Flat	Intermediate		Intermediate	Intermediate	Intermediate	
One-Click Updates	Yes	Yes (Add-on)	Yes	No	Yes (Add-on)	Limited (Add-on)	

[147] See http://www.cmsmatch.com/compare/content-management-systems/170+11+9+844+1462+43+1463

	Wordpress	**XOOPS**	**Plone**	**TYPO3**	**Joomla**	**Drupal**	**Alfresco**
Standards Compliance Level	High	High	High	High	High	High	
Site Setup Wizard	Yes	Yes	Limited	Yes	Limited (Paid)	Yes	Limited
Content Management							
Archives	Yes	Yes	Yes	Yes (Add-on)	Yes	Yes	
Content Categorization	Yes	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes	Yes	Yes
Content Construction Kit	Yes (Add-on)	Yes (Add-on)	Yes	Yes	Yes (Add-on)	Yes (Add-on)	
Content Staging and Merging	Yes (Add-on)	Yes	Yes (Add-on)	Yes	Yes (Add-on)	Limited (Add-on)	Yes
Content Tagging	Yes	Yes (Add-on)	Yes	Yes (Add-on)	Yes (Add-on)	Yes	Yes
Content Templates	Yes	Yes	Yes	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes
Custom Content Types	Yes	Yes	Yes	Yes (Add-on)	Yes (Add-on)	Yes	
Import-Export	Yes (Add-on)	Yes (Add-on)	Yes	Yes	Yes (Add-on)	Yes (Add-on)	Yes
Printer, Email and PDF Versions	Yes	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes	Yes (Add-on)	
Revisions and History	Yes	Yes (Add-on)	Yes	Yes	Limited (Add-on)	Yes	Yes
Scheduling	Yes	Yes (Add-on)	Yes	Yes	Yes	Yes (Add-on)	Yes
Subscriptions	Yes	Yes	Yes	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	No
Tag Cloud	Yes	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes
Voting and Rating	Yes (Add-on)	Yes	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Limited
Core Applications							
Advertising Management	Yes (Add-on)	Yes	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	No
Affiliate Tracking	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	No
Calendar	Yes	Yes (Add-on)	Yes	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes
Chat	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	No
Commenting System	Yes	Yes	Yes	Yes (Add-on)	Yes (Add-on)	Yes	
Contact Form	Yes	Yes	Yes	Yes	Yes	Yes	No
Contacts Management	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes	Yes (Add-on)	No

	Wordpress	**XOOPS**	**Plone**	**TYPO3**	**Joomla**	**Drupal**	**Alfresco**
Events Management	Yes (Add-on)	Yes (Add-on)	Yes	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	No
FAQ Management	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes
File Repository and Distribution	Yes (Add-on)	Yes (Add-on)	Yes	Yes	Yes (Add-on)	Yes (Add-on)	Yes
Forms and Surveys	Yes	Yes (Add-on)	Yes (Add-on)	Yes	Yes (Add-on)	Yes (Add-on)	No
Full-Text Document Search	Yes (Add-on)		Yes	Yes	Yes (Add-on)	Yes (Add-on)	
Graphs and Charts	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	No
Guestbook	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	No
Helpdesk / Ticketing System	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	No
HTTP Proxy		Yes	Yes (Add-on)	Yes		No	No
Internal Search Engine	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Lightbox (or variants)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes	Yes (Add-on)	
Link Management	Yes	Yes (Add-on)	Yes	Yes (Add-on)	Yes	Yes (Add-on)	Yes
Live Chat	Yes (Add-on)	Yes (Add-on)	Limited (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	
Media Gallery	Yes (Add-on)	Yes	Yes	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes
Newsletter Management	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	No
Polls	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes	Yes	No
Sitemap	Yes (Add-on)	Yes (Add-on)	Yes	Yes	Yes (Add-on)	Yes (Add-on)	Limited
Streaming Audio Management	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes	Yes (Add-on)	Yes (Add-on)	
Streaming Video Management	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes	Yes (Add-on)	Yes (Add-on)	
Tests, Quizzes and Raffles	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	No
XML Sitemap for Search Engines	Yes (Add-on)	Yes (Add-on)	Yes	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	
Flexibility							
Interface Localization (l10n)	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Internationalization (i18n)	Limited	Yes	Yes	Yes	Yes (Add-on)	Yes	Yes
Language Negotiation	Limited	Yes	Yes	Yes	Yes (Add-	Yes	

	Wordpress	**XOOPS**	**Plone**	**TYPO3**	**Joomla**	**Drupal**	**Alfresco**
					on)		
Multi-Site from 1 Codebase	Yes (Add-on)	Yes (Add-on)	Yes	Yes	Yes (Paid)	Yes	Yes
Multi-Site from 1 Database	Yes (Add-on)	Yes (Add-on)	Yes	Yes	Yes (Paid)	Yes	
Multiple Domains Management	Yes (Add-on)	Yes	Yes	Yes	Limited (Add-on)	Yes (Add-on)	Yes
Right to Left Language Support	Yes	Yes	Yes	Yes (Add-on)	Yes	Yes	
Interoperability							
CGI Mode Support	No	Yes	Yes	Yes	No	Yes (Add-on)	No
Content Syndication (RSS)	Yes	Yes	Yes	Yes (Add-on)	Yes	Yes	Yes
Database Query Editor	Yes (Add-on)	Yes (Add-on)	Yes	Yes	Yes (Add-on)	Yes (Add-on)	
Database-to-Web External Databases	Limited (Add-on)	Yes	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes	
FTP Support	Limited (Add-on)	Yes (Add-on)	Yes	Yes (Add-on)	Yes	Yes (Add-on)	Yes
iCal	Yes (Add-on)	Yes (Add-on)	Yes	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	No
Section 508	Yes	Limited	Yes	Yes	Limited	Yes (Add-on)	
Text Browser Support	Yes (Add-on)	Yes	Yes	Yes	Yes (Add-on)	Yes (Add-on)	
UTF-8 Support	Yes	Yes	Yes	Yes	Yes	Yes	Yes
W3C XHTML Compliant	Yes	Yes	Yes	Yes	Yes	Yes	Yes
WAI Compliant	Limited	Limited	Yes	Yes	Limited	Yes (Add-on)	Yes
Web Services API	Yes (Add-on)	Yes	Yes	Yes (Add-on)	Yes (Add-on)	Yes	Yes
WebDAV Support	Yes (Add-on)	Yes (Add-on)	Yes	Limited (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes
Layouts, Design and Templates							
Content Type Theming	Yes (Add-on)	Yes	Yes	Yes	Yes (Add-on)	Yes	
Drag and Drop Layouts	Yes (Add-on)		No	Yes (Add-on)	No	Yes (Add-on)	
Granular CSS Classes	Yes (Add-on)	Yes	Yes	Yes	Yes	Yes	
Scheduled Theming	Yes (Add-on)	Yes (Add-on)		Yes	No	Yes (Add-on)	
Style Wizard	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Limited	No
Sub Theming	Yes (Add-	Yes	Yes	Yes (Add-	Yes	Yes	

	Wordpress	**XOOPS**	**Plone**	**TYPO3**	**Joomla**	**Drupal**	**Alfresco**
	on)		(Add-on)	on)			
Template Language	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Theming / Skinning	Yes	Yes	Yes	Yes	Yes	Yes	Yes
URL Path Theming	Yes (Add-on)	Yes	Yes (Add-on)	Yes	Yes	Yes	
Web-based Template Management	Yes	Yes	Yes	Yes	Limited	Yes	Yes
Mobile Internet							
Device Capabilities Caching	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	No	Limited (Add-on)	Limited (Add-on)	
Device Capabilities Detection	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Limited	Limited (Add-on)	Yes (Add-on)	
Device Groups	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)		No	Limited (Add-on)	
Site Wizard for Mobile Site	Yes (Add-on)	Yes (Add-on)			No	Limited (Add-on)	
SMS Support	Yes (Add-on)		Limited (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	
Templates per Device Group	Yes (Add-on)	Yes (Add-on)	Yes		Limited (Add-on)	Limited (Add-on)	
Unique Mobile Content	Yes (Add-on)	Yes (Add-on)	Limited (Add-on)	Yes	Limited (Add-on)	Yes (Add-on)	
Page Editing							
Clipboard	Yes	Yes	Yes	Yes	Yes	No	Yes
Copy / Paste from Office	Yes	Yes	Yes	Yes	Limited (Add-on)	Yes (Add-on)	Yes
Email Content to Site	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes
External Editor	No	Yes (Add-on)	Yes (Add-on)	Limited (Add-on)	Yes	Limited (Add-on)	Limited
Image Auto Thumbnails	Yes	Yes	Yes	Yes	Yes (Add-on)	Yes (Add-on)	Yes
Image Editing	Yes	Yes	Yes (Add-on)	Yes (Add-on)	Limited (Add-on)	Yes (Add-on)	Limited (Add-on)
Image Resizing	Yes	Yes	Yes	Yes	Yes (Add-on)	Yes (Add-on)	Yes
Macro Language	No	Yes	Yes	Yes	No	Limited	Yes
Server Page Language	Yes (Add-on)	Yes	Yes	Yes	Yes (Add-on)	Yes	Yes
Spelling Checker	Yes	Yes (Add-on)	Yes (Add-on)	Yes	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)
Trash Bin	Yes	Yes	Yes (Add-on)	Yes	Yes	Yes (Add-on)	Yes
Undo History	Yes	Yes (Add-on)	Yes	Yes	Yes	No	Yes
WYSIWYG Editor	Yes	Yes	Yes	Yes	Yes	Yes (Add-on)	Yes

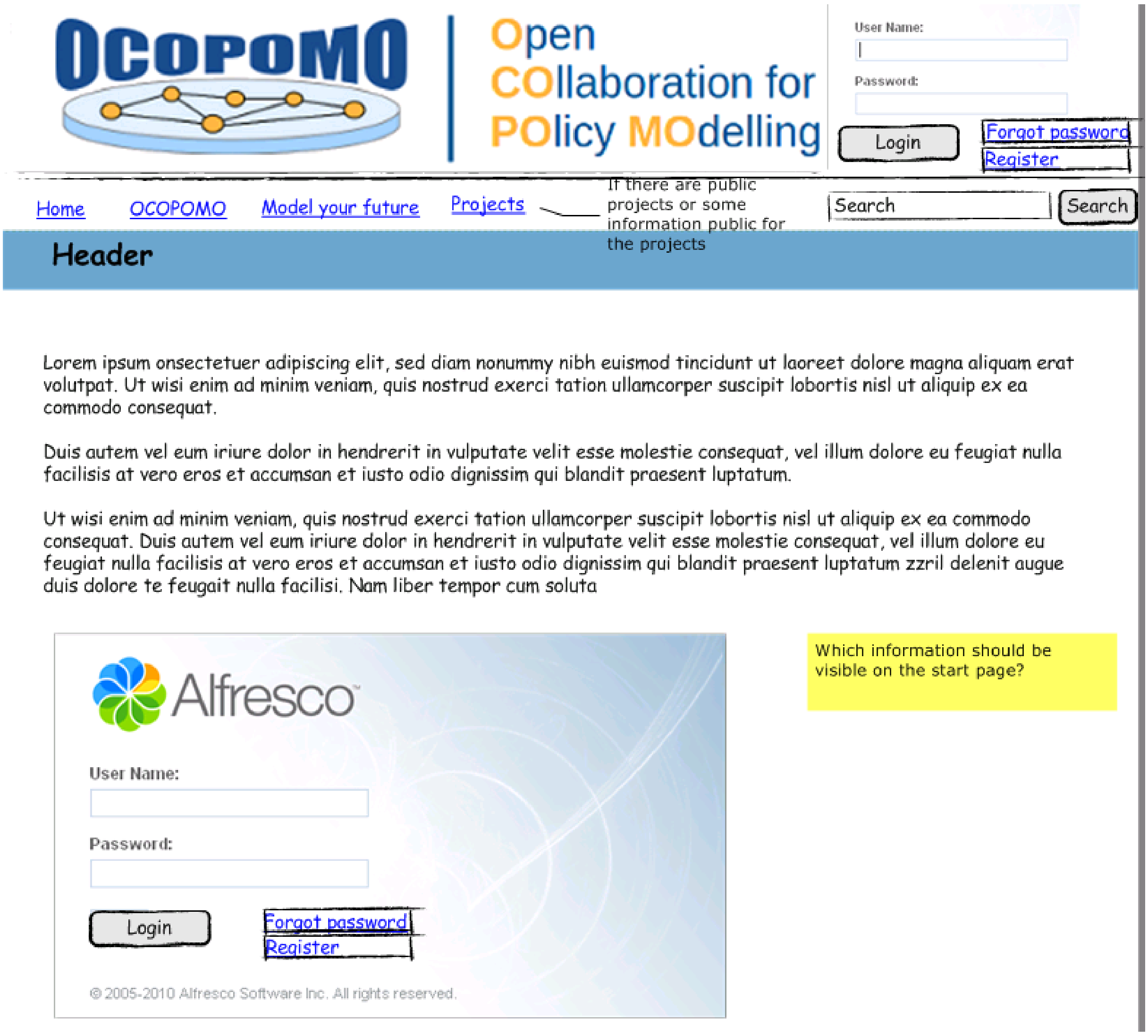
	Wordpress	**XOOPS**	**Plone**	**TYPO3**	**Joomla**	**Drupal**	**Alfresco**
Performance							
Advanced Caching	Yes (Add-on)	Yes	Yes	Yes	Yes	Yes	Yes
Automatic Meta Tags	Yes (Add-on)	Yes (Add-on)	Yes	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes
Bandwidth Optimization	Yes (Add-on)	Yes	Yes	Yes (Add-on)	Limited (Add-on)	Yes	Limited
Database Optimization	Yes (Add-on)	Yes (Add-on)	Yes	Yes	Limited (Add-on)	Yes (Add-on)	No
Database Query Caching	Limited	Yes	Yes	Yes	Yes (Add-on)	Yes	
Database Replication	Yes (Add-on)	No	Yes	Yes	Yes (Add-on)	Yes (Add-on)	Yes
Friendly URLs	Yes	Yes (Add-on)	Yes	Yes (Add-on)	Yes	Yes	Yes
Load Balancing	Yes (Add-on)	Yes	Yes	Yes	Limited (Add-on)	Yes (Add-on)	Yes
Minify Javascipt	Yes (Add-on)	Yes (Add-on)	Yes	Yes (Add-on)	Limited (Add-on)	Yes	
Search Engine Optimization	Yes	Yes (Add-on)	Yes	Yes	Yes (Add-on)	Yes (Add-on)	Yes
Static Content Export	Yes	Yes	Yes (Add-on)	Yes	Yes	Yes (Add-on)	Yes
Security							
Audit Trail	Limited	Yes (Add-on)	Yes	Yes	Limited	Yes	Yes
Captcha Anti-Spam	Yes	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	No
Content Approval	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Database Backup/Restore	Yes (Add-on)	Yes (Add-on)	Yes	Yes	Yes (Add-on)	Yes (Add-on)	Yes
Email Verification	Yes	Yes	Yes	Yes	Yes	Yes	No
Error Reporting	Limited	Limited	Yes (Add-on)	Yes	Limited	Yes	Yes
Kerberos Authentication		No	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes
LDAP Authentication		Yes (Add-on)	Yes	Yes (Add-on)	Yes	Yes (Add-on)	Yes
Login History	Limited	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Limited	Yes	Yes
NIS Authentication		Yes (Add-on)	Yes (Add-on)	Yes (Add-on)		Limited (Add-on)	No
NTLM Authentication		Yes (Add-on)	Yes (Add-on)	Yes (Add-on)		Yes (Add-on)	Yes
Password Encryption	Yes	Yes	Yes	Yes	Yes	Yes	
Pluggable Authentication	Limited	Yes (Add-on)	Yes	Yes	Yes	Yes (Add-on)	Yes

	Wordpress	**XOOPS**	**Plone**	**TYPO3**	**Joomla**	**Drupal**	**Alfresco**
Sandbox	Yes (Add-on)	Yes	Yes	Yes	Yes (Add-on)	Yes (Add-on)	Yes
Scheduled Backups	Limited (Add-on)	Yes (Add-on)	Yes (Add-on)	Limited (Add-on)	Yes (Add-on)	Yes (Add-on)	
Session Management		Yes	Yes	Yes	Yes	Yes	Limited
SMB Authentication		No	Yes (Add-on)	Yes		No	Yes
SSL Support	Yes (Add-on)	Yes	Yes	Yes	Yes (Add-on)	Yes (Add-on)	Yes
Versioning	Yes	Yes	Yes	Yes	Yes (Add-on)	Yes	Yes
Site Administration							
Administration Dashboard	Yes	Yes (Add-on)	Yes	Yes	Yes	Limited	Yes
Drag and Drop Interface	Yes	Yes (Add-on)	Yes	Limited	No	Yes	Limited
File and Document Manager	Yes (Add-on)	Yes (Add-on)	Yes	Yes	Yes (Add-on)	Yes (Add-on)	Yes
Google Analytics	Yes (Add-on)	Yes (Add-on)	Yes	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	
Inline Content Administration	Yes (Add-on)	Yes	Yes	Yes	Limited	Yes	No
Internal Search for Admin	Yes	Yes	Yes	Yes	Yes	Yes	
Link Checker	Yes (Add-on)	Limited	Yes	Yes	Yes (Add-on)	Yes (Add-on)	Limited
Mass Upload	Yes	Yes	Yes	Yes	Yes (Add-on)	Yes (Add-on)	Yes
Media Library	Yes	Yes	Yes	Yes (Add-on)	Yes	Yes (Add-on)	Yes
Metadata	Yes	Yes	Yes	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes
Off-line Maintenance Page	Yes	Yes	No	Yes	Yes	Yes	
On-line Administration	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Personal Dashboard	Yes	Yes (Add-on)	Yes	Yes	Limited	Yes (Add-on)	
Site Navigation Management	Yes (Add-on)	Yes	Yes	Yes	Yes	Yes	Yes
Statistics	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes	Yes	No
Translation Strings Management		Yes (Add-on)	Yes	Yes	Yes (Add-on)	Yes (Add-on)	Yes
Workflow Engine	Yes (Add-on)	Yes (Add-on)	Yes	Yes	Limited	Yes (Add-on)	Yes
Zip Archive Support	Yes	Yes	Yes	Yes (Add-	Yes (Add-	Yes (Add-	Yes

	Wordpress	**XOOPS**	**Plone**	**TYPO3**	**Joomla**	**Drupal**	**Alfresco**
			(Add-on)	on)	on)	on)	
Support							
Certification Programme	No	Yes	No	Yes	No	Yes	Yes
Code Skeletons	Limited	Yes	Yes	Yes	Limited	Yes	Yes
Commercial Manuals	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Commercial Support	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Community Forums	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Developer Community	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Issue Tracking	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Mailing Lists	Yes	Yes	Yes	Yes	Yes	Yes	No
On-line Help	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Support Network Subscription	Yes (Paid)	Yes (Add-on)	Yes (Paid)		Yes (Paid)	Yes (Add-on)	
Trainings and Seminars	Yes	Yes	Yes	Yes	Yes	Yes	Yes
User Conferences	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Users							
Avatars		Yes	Limited	Yes (Add-on)	Yes	Yes (Add-on)	
Buddy List	Yes (Add-on)	Yes (Add-on)		Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	
Memberlist		Yes	Yes	Yes (Add-on)	Yes	Yes (Add-on)	
Memberlist Search		Yes	Yes	Yes (Add-on)	Yes	Yes	
OpenID Login Support	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes	Yes	Yes (Add-on)	Limited
Paid Content Subscriptions	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	No
Private Messaging System	Yes (Add-on)	Yes	Limited (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	No
Public User Page	Yes (Add-on)	Yes (Add-on)	Yes	Yes (Add-on)	Yes (Paid)	Yes	
Registration Form	Yes (Add-on)	Yes	Yes	Yes (Add-on)	Yes	Yes	Yes
Registration Form Custom Fields	Yes (Add-on)	Yes	Yes	Yes (Add-on)	Yes (Add-on)	Yes	Limited
User Access Control	Yes (Add-on)	Yes	Yes	Yes	Yes (Add-on)	Yes	Yes
User Contributions	Yes (Add-on)	Yes	Yes	Yes (Add-on)	Yes	Yes	Limited
User Groups	Yes (Add-on)	Yes	Yes	Yes	Yes (Add-on)	Yes (Add-on)	Yes
User Points / Karma Rating		Yes (Add-on)	Limited (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	

	Wordpress	**XOOPS**	**Plone**	**TYPO3**	**Joomla**	**Drupal**	**Alfresco**
User Preferences	Yes	Yes	Limited (Add-on)	Yes (Add-on)	Yes	Yes	
User Profile Custom Fields	Yes (Add-on)	Yes	Yes (Add-on)	Yes (Add-on)	Yes (Add-on)	Yes	
User Profiles	Yes	Yes	Yes	Yes (Add-on)	Yes (Add-on)	Yes	Yes
User Signatures		Yes		Yes (Add-on)		Yes (Add-on)	
Who's On-line List		Yes		Yes (Add-on)	Yes (Add-on)	Yes	

APPENDIX C: USER INTERFACE MOCK-UPS



If there are public projects or some information public for the projects

Lorem ipsum onsectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

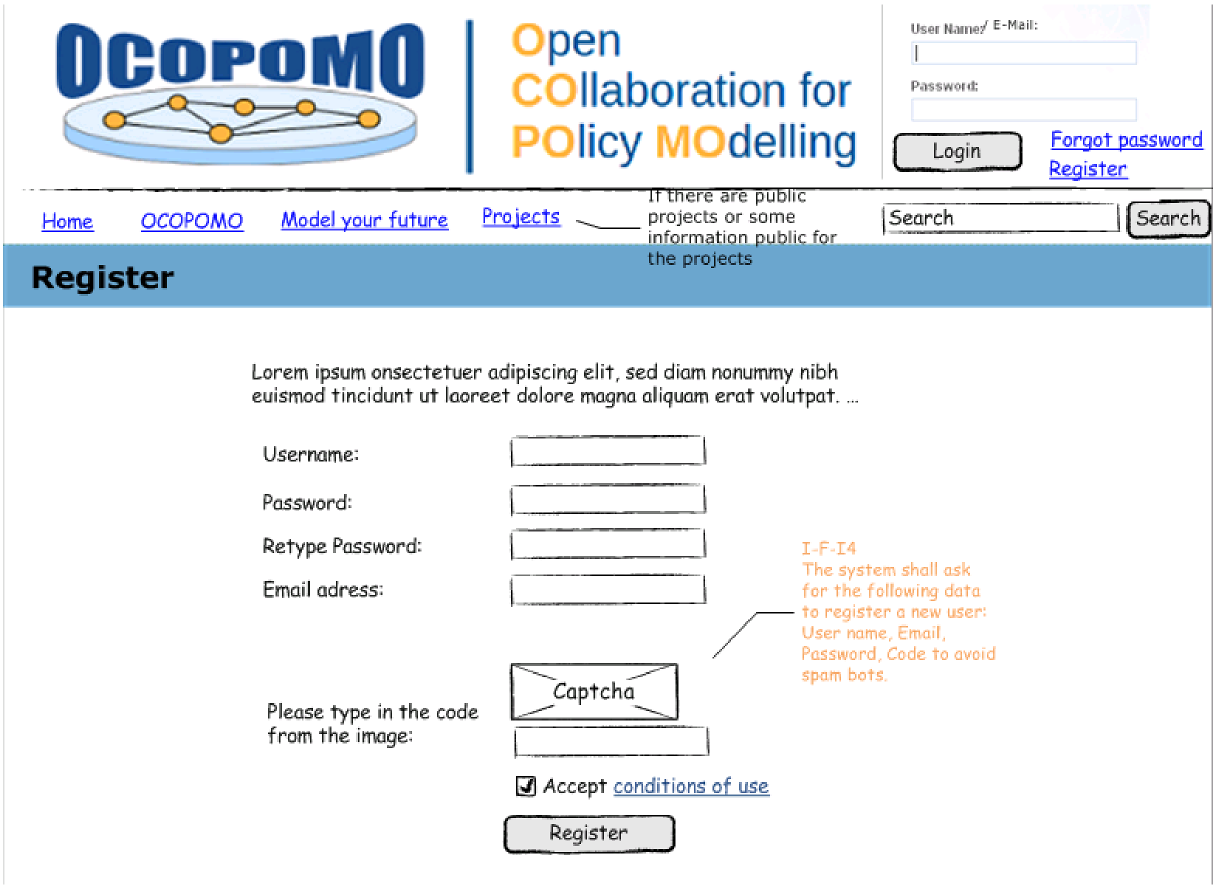
Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum.

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta

© 2005-2010 Alfresco Software Inc. All rights reserved.

Which information should be visible on the start page?

Figure 63 Mock-up for Home page



Lorem ipsum onsectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. ...

Username:

Password:

Retype Password:

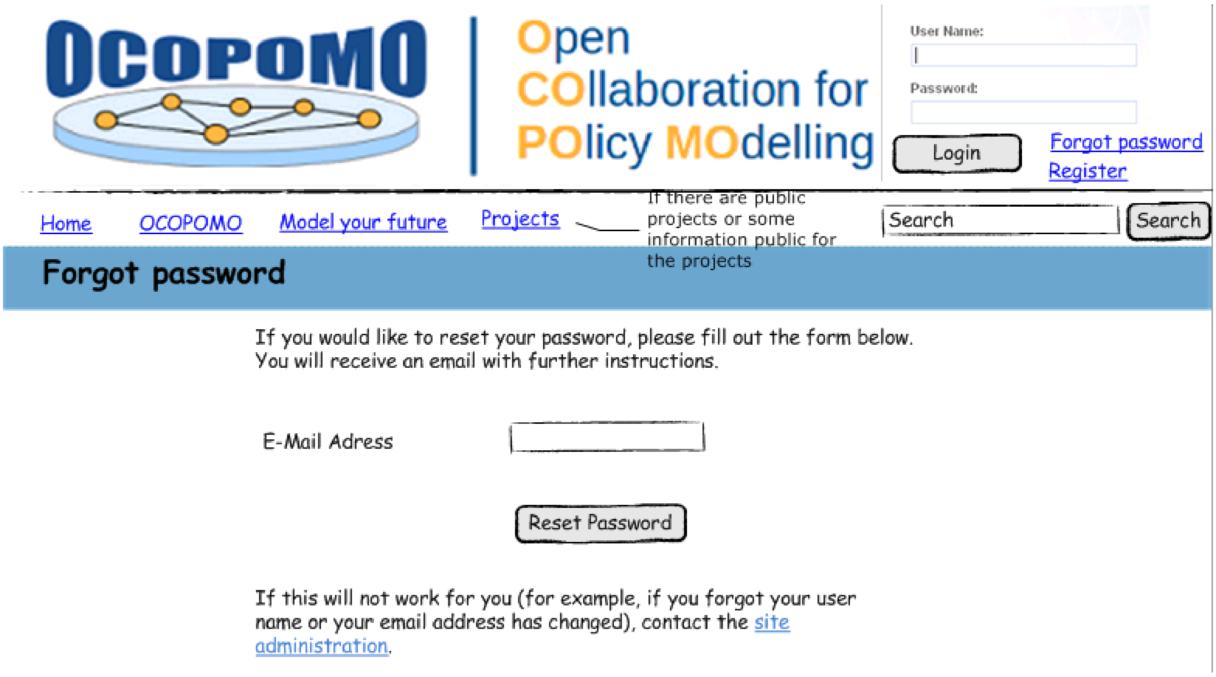
Email adress:

Please type in the code from the image:

☑ Accept conditions of use

I-F-I4
The system shall ask for the following data to register a new user: User name, Email, Password, Code to avoid spam bots.

Figure 64 Mock-up for registration at the system

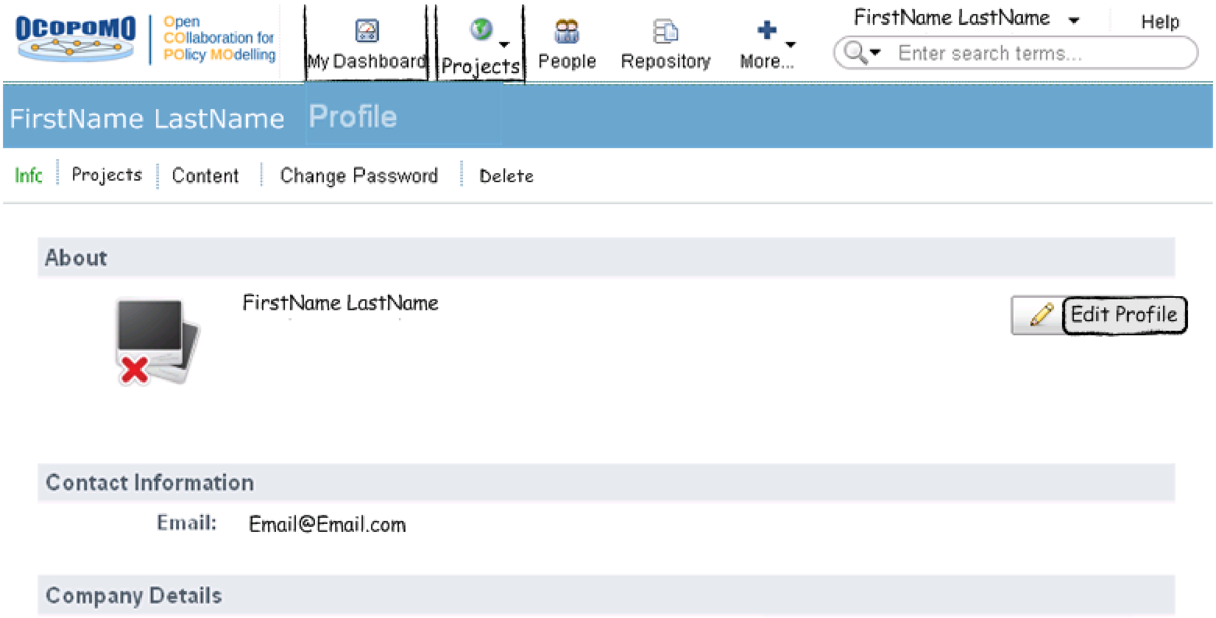


If you would like to reset your password, please fill out the form below. You will receive an email with further instructions.

E-Mail Adress

If this will not work for you (for example, if you forgot your user name or your email address has changed), contact the site administration.

Figure 65 Mock-up for password prompt



About

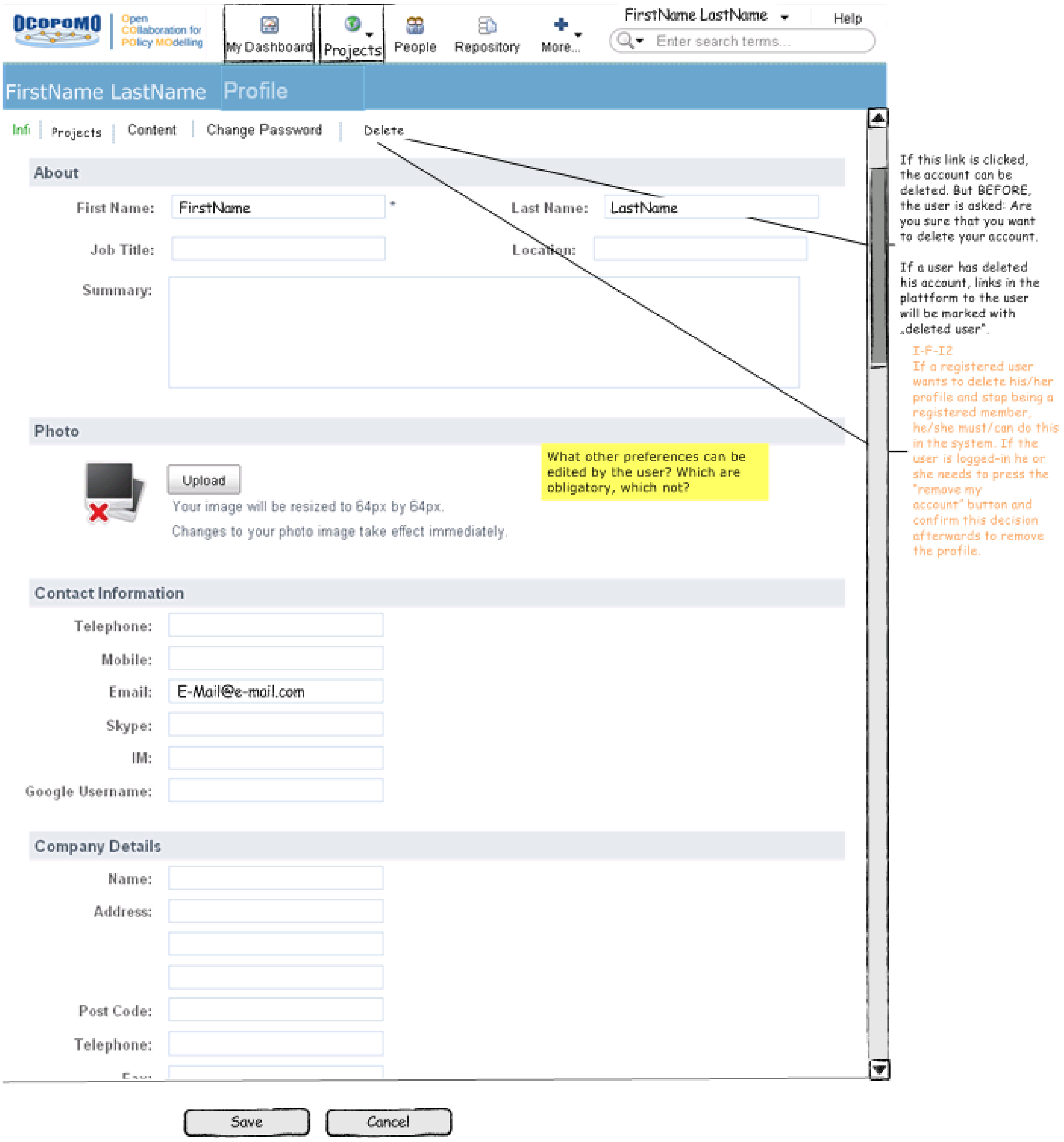
FirstName LastName

Contact Information

Email: Email@Email.com

Company Details

Figure 66 Mock-up for user’s profile



First Name: FirstName *

Last Name: LastName

Job Title:

Location:

Summary:

Your image will be resized to 64px by 64px.

Changes to your photo image take effect immediately.

What other preferences can be edited by the user? Which are obligatory, which not?

Telephone:

Mobile:

Email: E-Mail@e-mail.com

Skype:

IM:

Google Username:

Name:

Address:

Post Code:

Telephone:

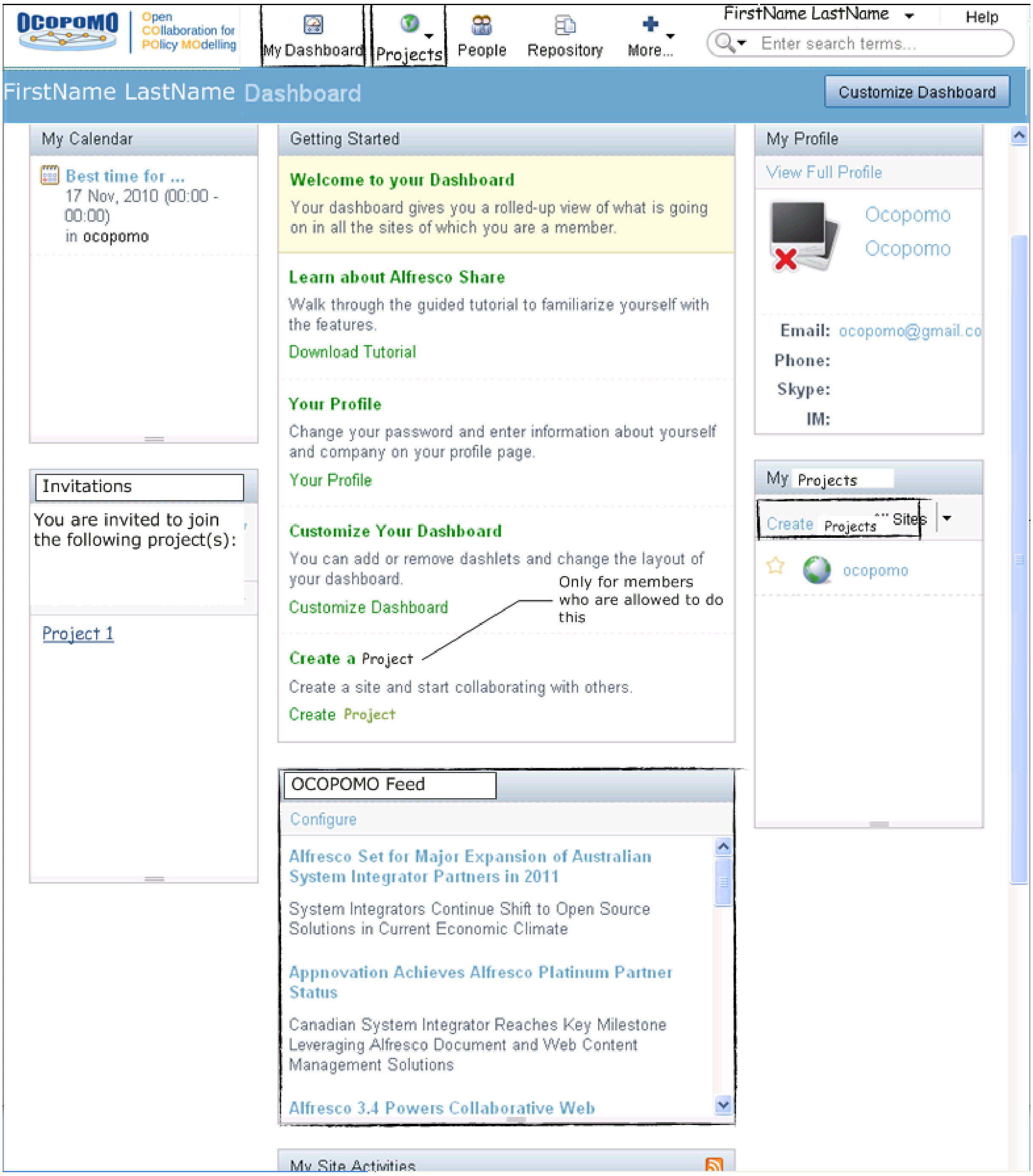
Fax:

If this link is clicked, the account can be deleted. But BEFORE, the user is asked: Are you sure that you want to delete your account.

If a user has deleted his account, links in the plattform to the user will be marked with „deleted user".

I-F-I2
If a registered user wants to delete his/her profile and stop being a registered member, he/she must/can do this in the system. If the user is logged-in he or she needs to press the "remove my account" button and confirm this decision afterwards to remove the profile.

Figure 67 Mock-up for editing user's profile



FirstName LastName Dashboard

My Calendar

Best time for ...
17 Nov, 2010 (00:00 - 00:00)
in ocopomo

Invitations

You are invited to join the following project(s):

Project 1

Getting Started

Welcome to your Dashboard

Your dashboard gives you a rolled-up view of what is going on in all the sites of which you are a member.

Learn about Alfresco Share

Walk through the guided tutorial to familiarize yourself with the features.

Download Tutorial

Your Profile

Change your password and enter information about yourself and company on your profile page.

Your Profile

Customize Your Dashboard

You can add or remove dashlets and change the layout of your dashboard.

Customize Dashboard

Only for members who are allowed to do this

Create a Project

Create a site and start collaborating with others.

Create Project

OCOPOMO Feed

Alfresco Set for Major Expansion of Australian System Integrator Partners in 2011

System Integrators Continue Shift to Open Source Solutions in Current Economic Climate

Appnovation Achieves Alfresco Platinum Partner Status

Canadian System Integrator Reaches Key Milestone Leveraging Alfresco Document and Web Content Management Solutions

Alfresco 3.4 Powers Collaborative Web

My Site Activities

My Profile

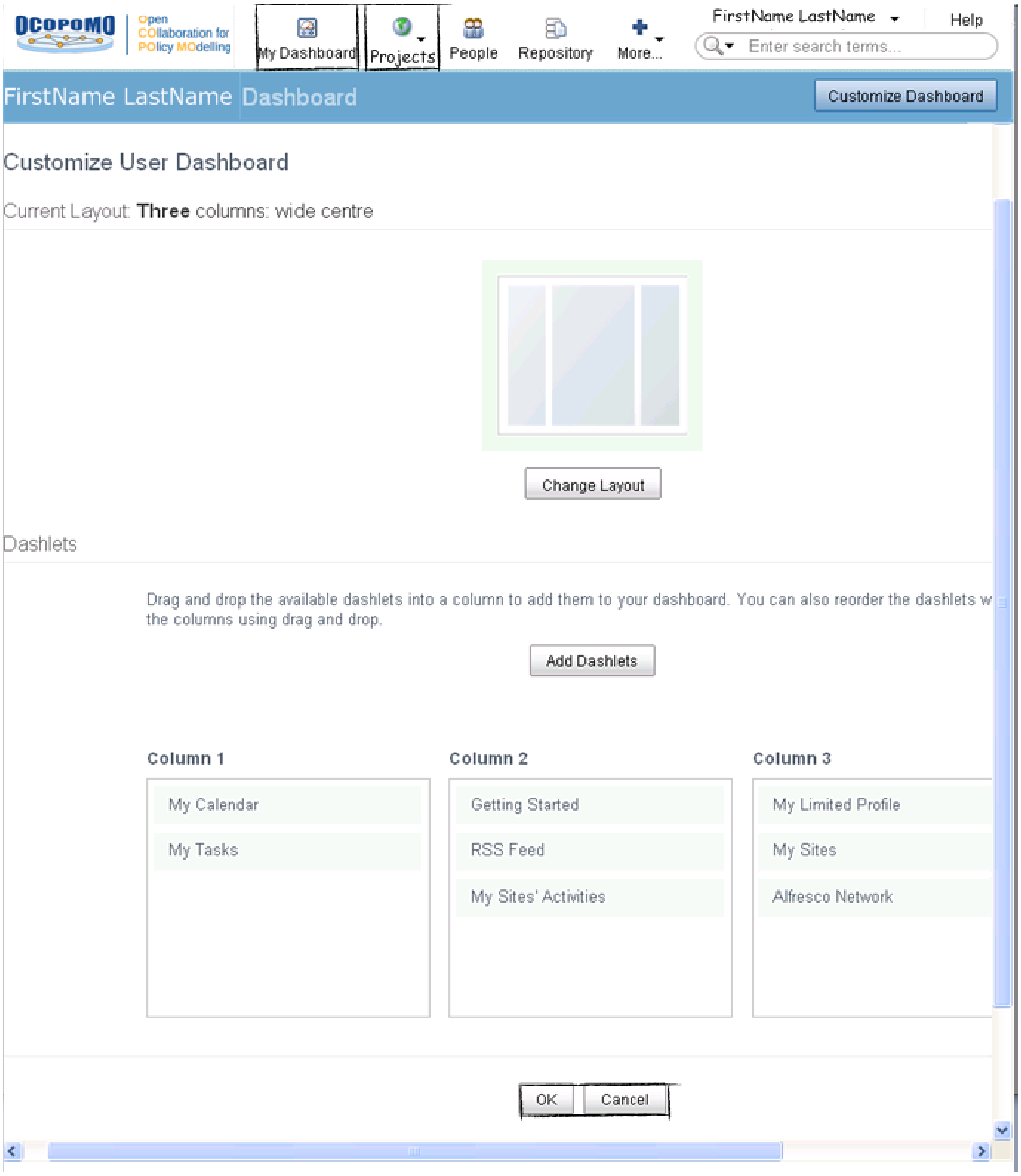
Ocopomo
Ocopomo

Email: ocopomo@gmail.co
Phone:
Skype:
IM:

My Projects

ocopomo

Figure 68 Mock-up for the Dashboard



Customize User Dashboard

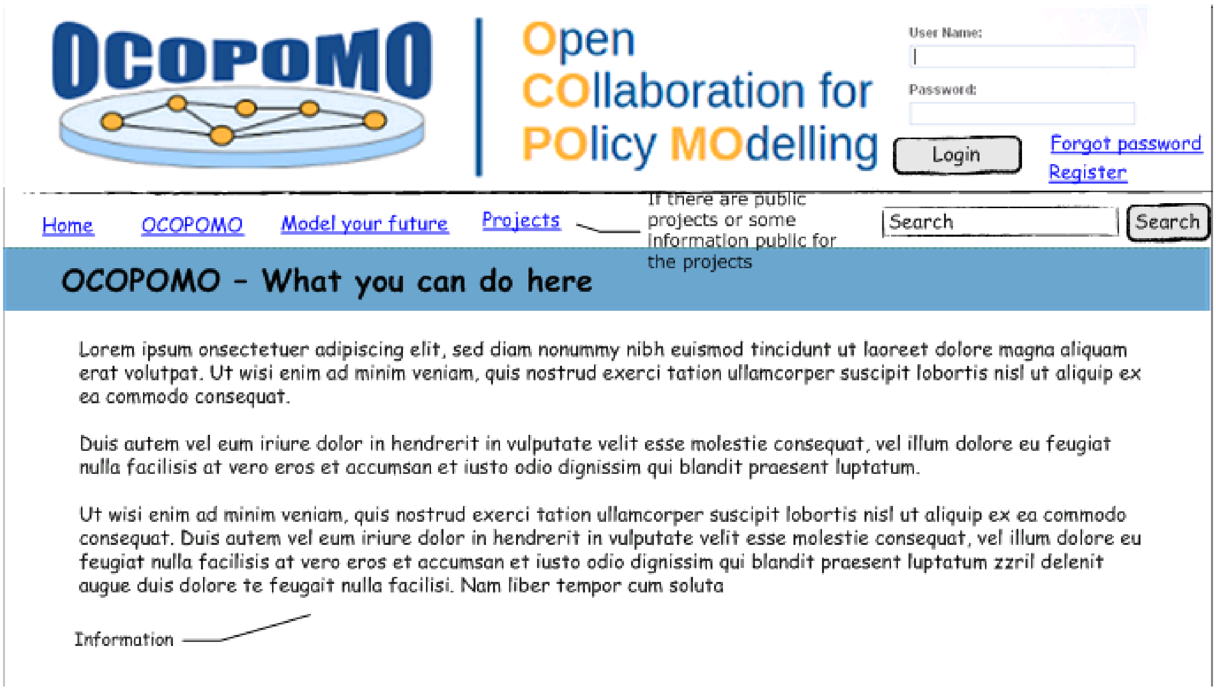
Current Layout: **Three** columns: wide centre

Dashlets

Drag and drop the available dashlets into a column to add them to your dashboard. You can also reorder the dashlets w the columns using drag and drop.

Column 1	Column 2	Column 3
My Calendar	Getting Started	My Limited Profile
My Tasks	RSS Feed	My Sites
	My Sites' Activities	Alfresco Network

Figure 69 Mock-up for customising the Dashboard



If there are public projects or some information public for the projects

OCOPOMO - What you can do here

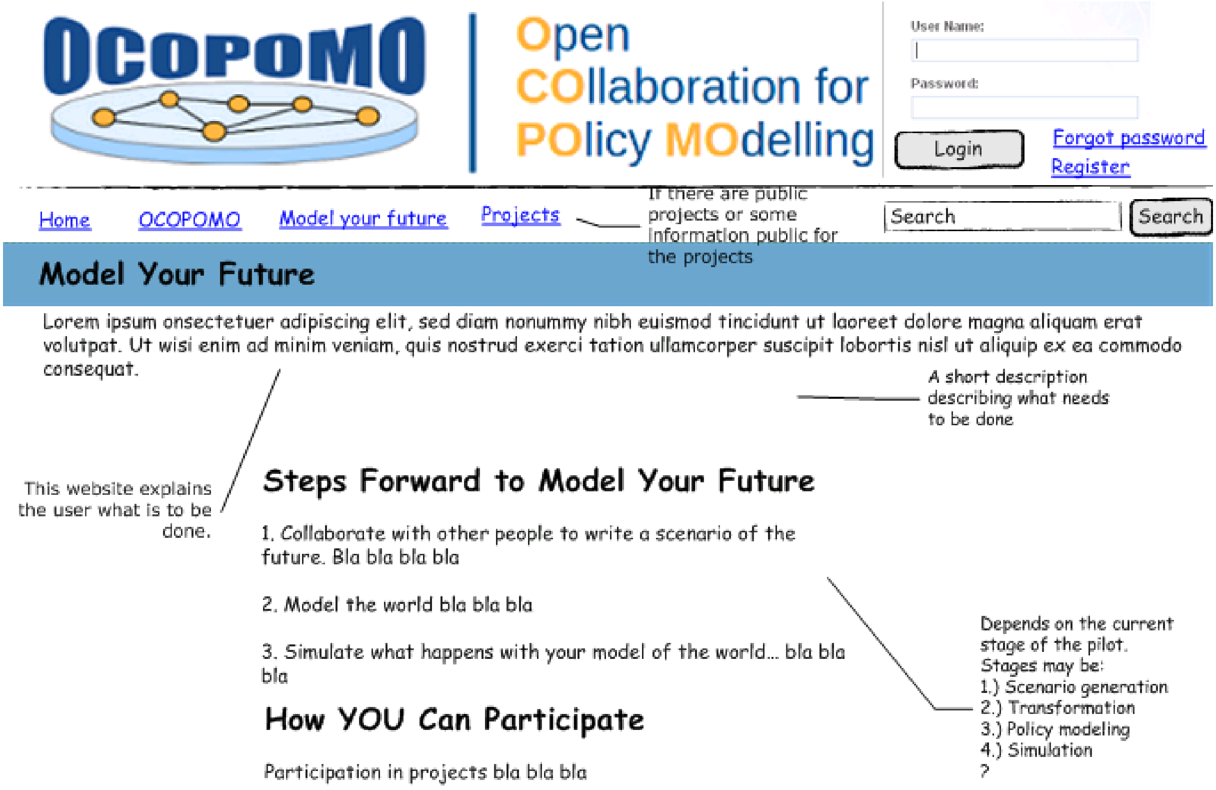
Lorem ipsum onsectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum.

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta

Information

Figure 70 Mock-up that shows the OCOPOMO project description



If there are public projects or some information public for the projects

Model Your Future

Lorem ipsum onsectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

A short description describing what needs to be done

This website explains the user what is to be done.

Steps Forward to Model Your Future

1. Collaborate with other people to write a scenario of the future. Bla bla bla bla

2. Model the world bla bla bla

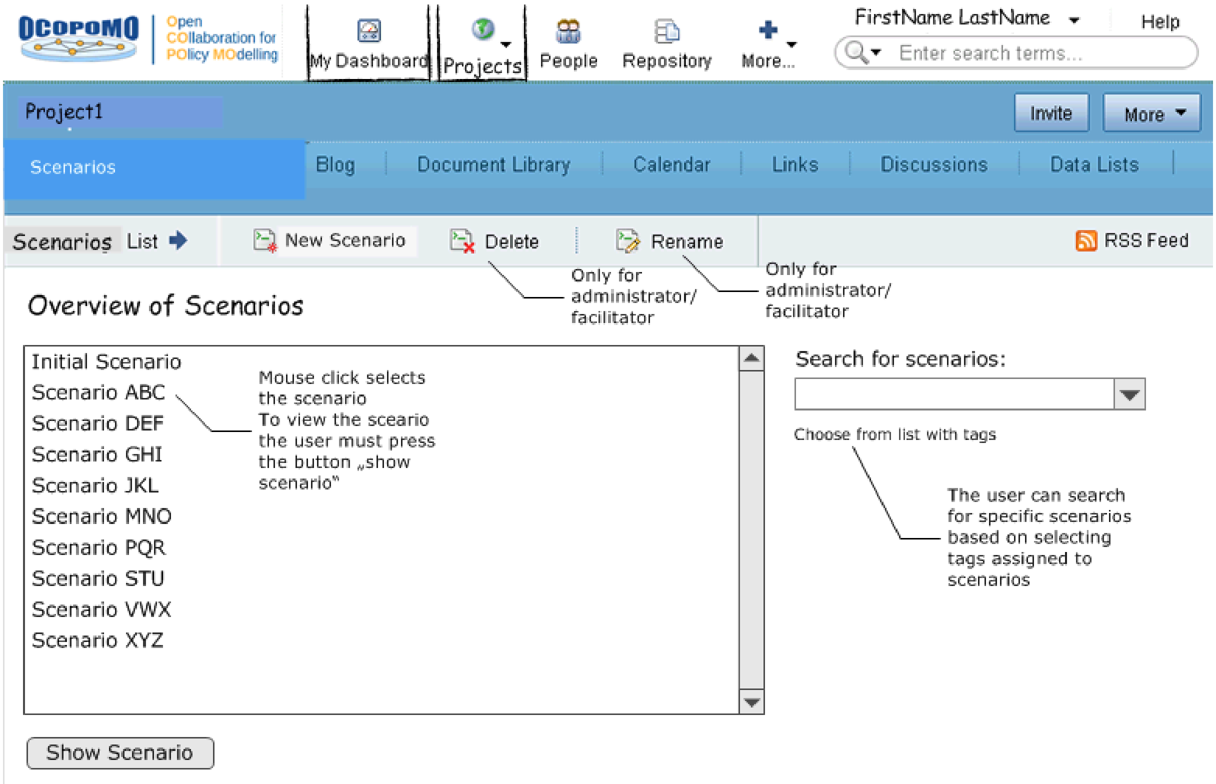
3. Simulate what happens with your model of the world… bla bla bla

How YOU Can Participate

Participation in projects bla bla bla

Depends on the current stage of the pilot.
Stages may be:
1.) Scenario generation
2.) Transformation
3.) Policy modeling
4.) Simulation
?

Figure 71 Mock-up that shows where users start to contribute



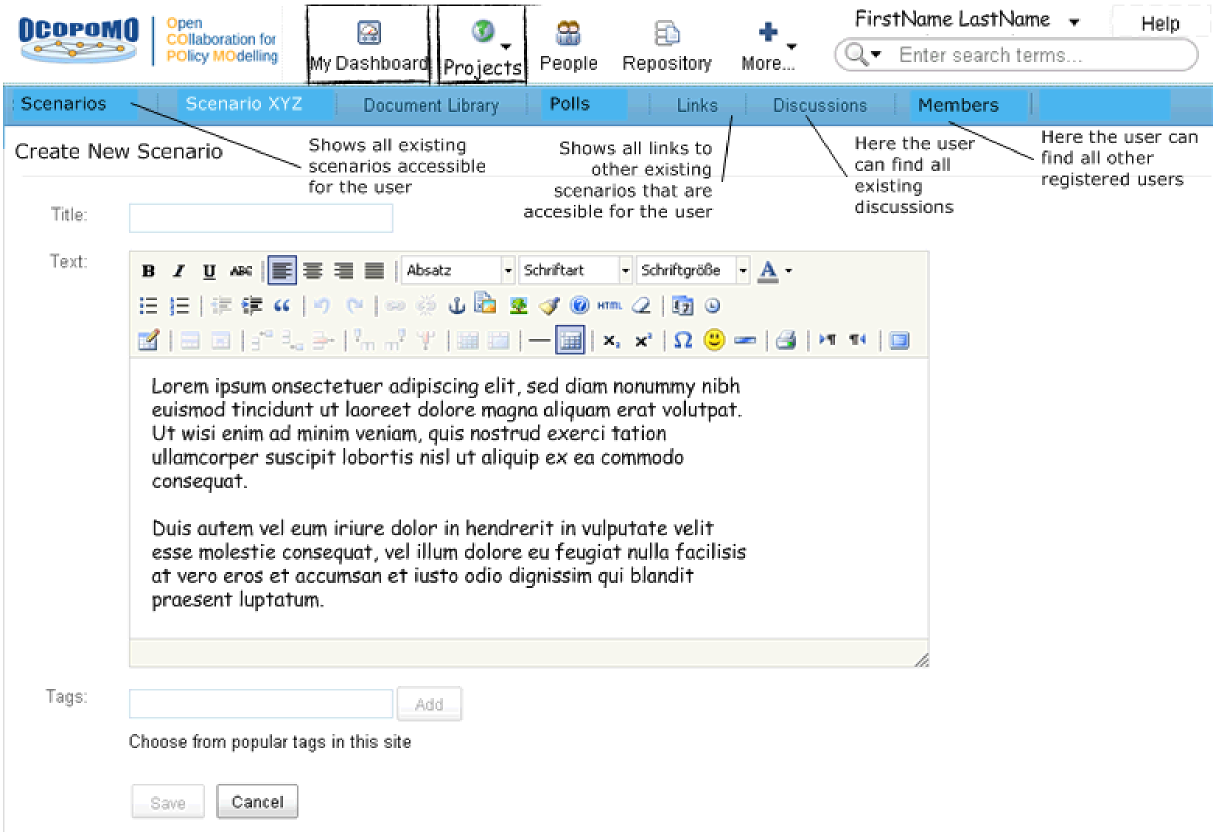
Only for administrator/ facilitator

Only for administrator/ facilitator

Mouse click selects the scenario
To view the sceario the user must press the button „show scenario"

The user can search for specific scenarios based on selecting tags assigned to scenarios

Figure 72 Mock-up for starting with collaborative scenario building by viewing existing ones



Shows all existing scenarios accessible for the user

Shows all links to other existing scenarios that are accesible for the user

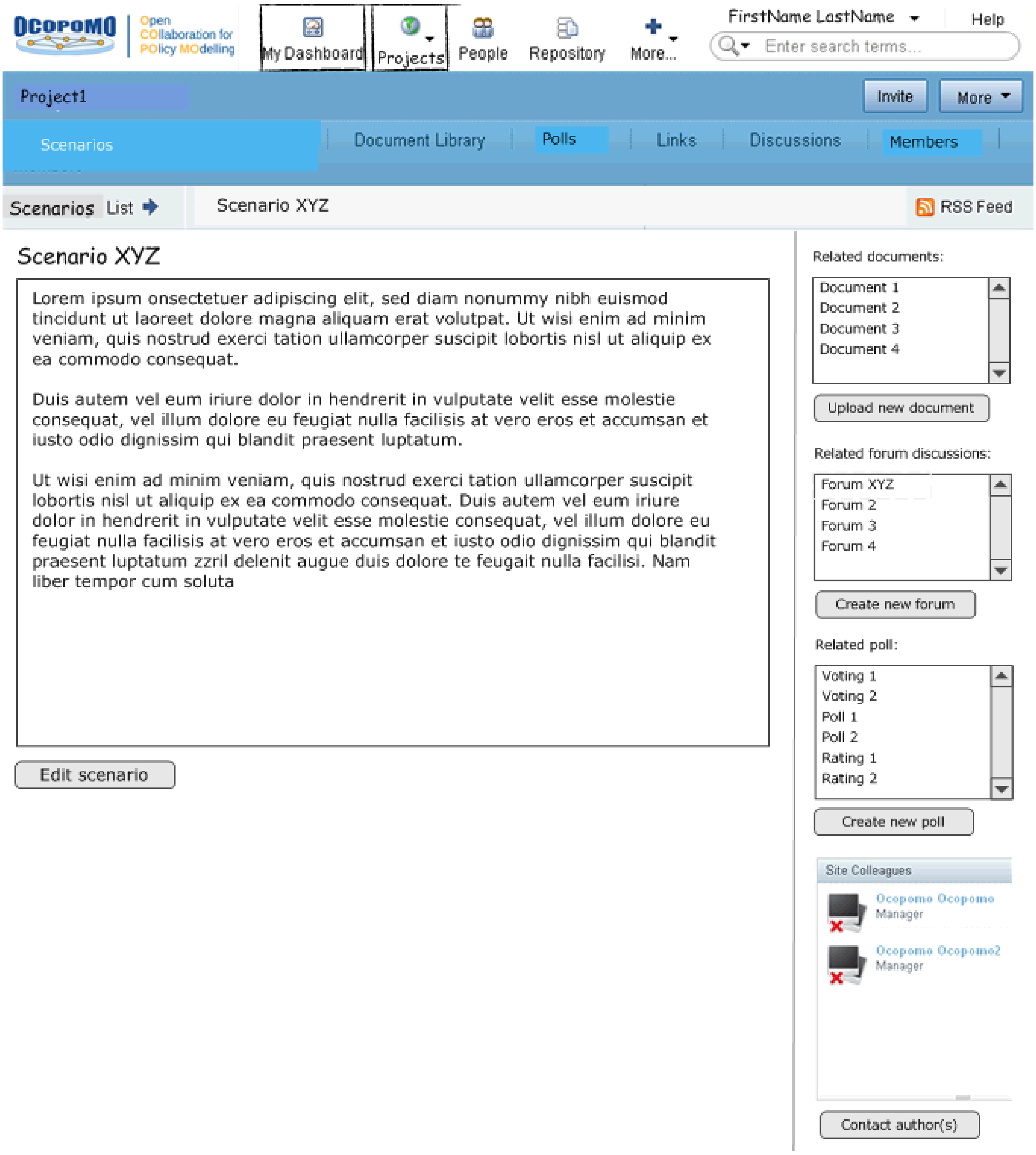
Here the user can find all existing discussions

Here the user can find all other registered users

Lorem ipsum onsectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum.

Figure 73 Mock-up for creating a new scenario

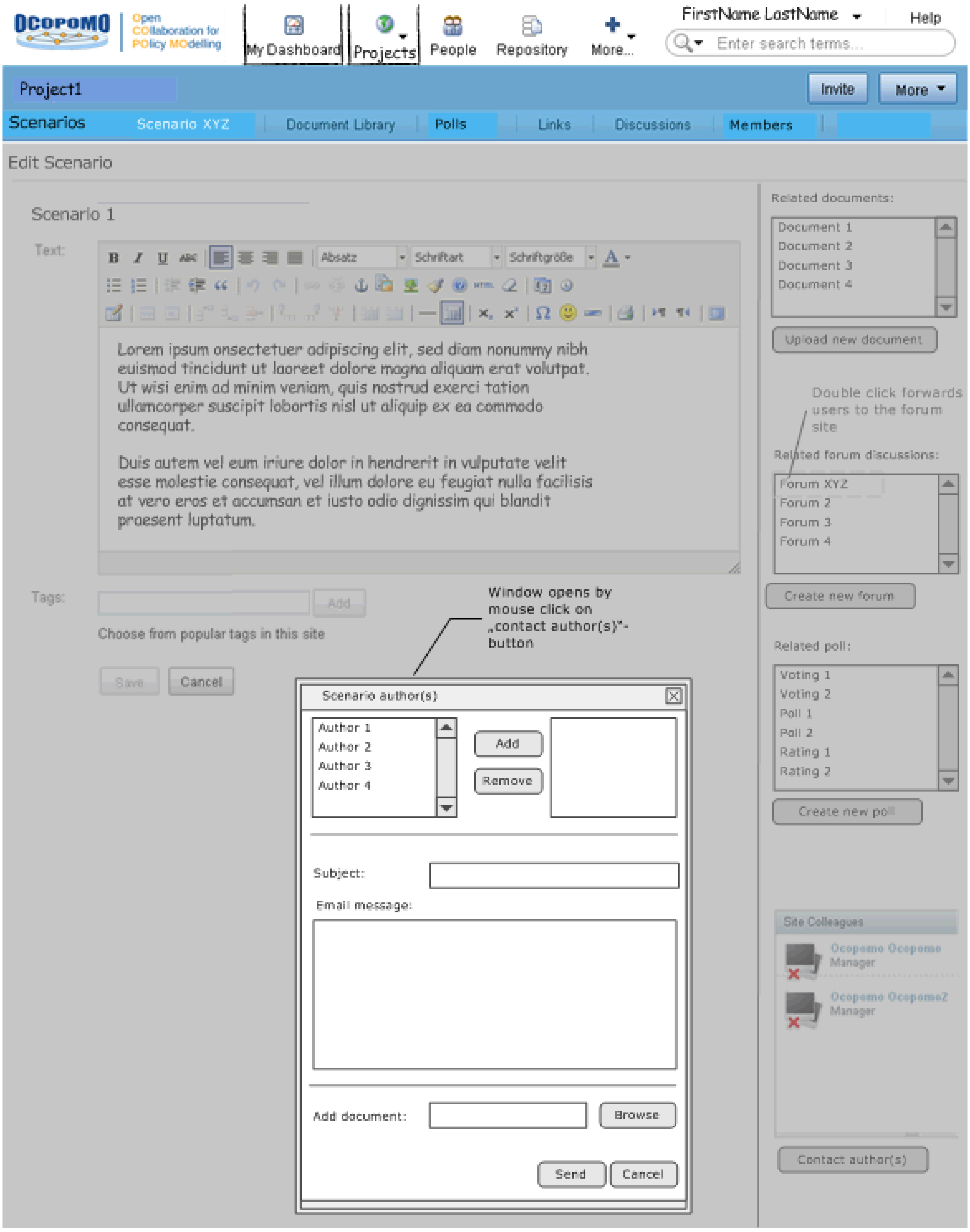


Lorem ipsum onsectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum.

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta

Figure 74 Mock-up for viewing scenarios



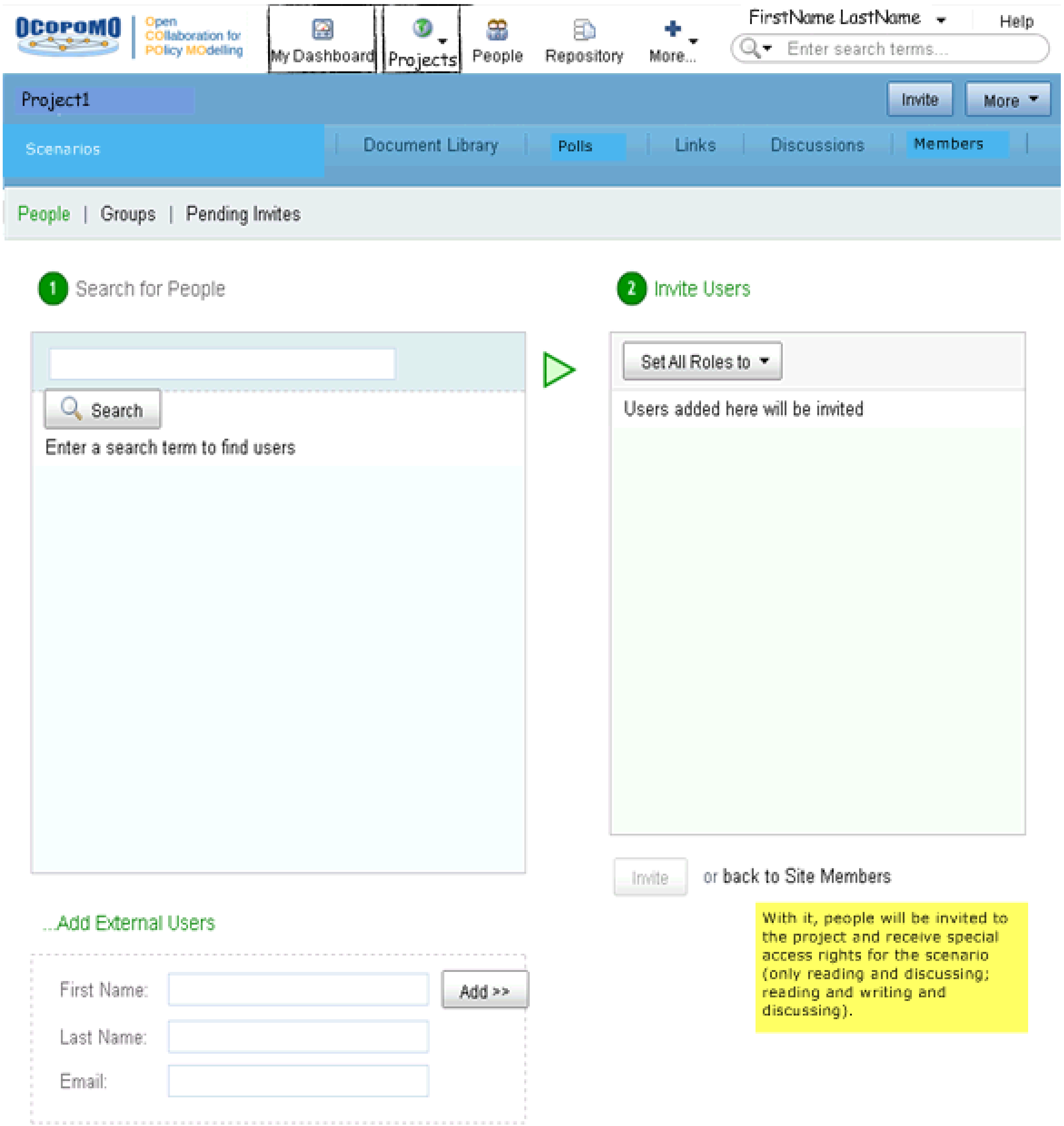
Lorem ipsum onsectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum.

Double click forwards users to the forum site

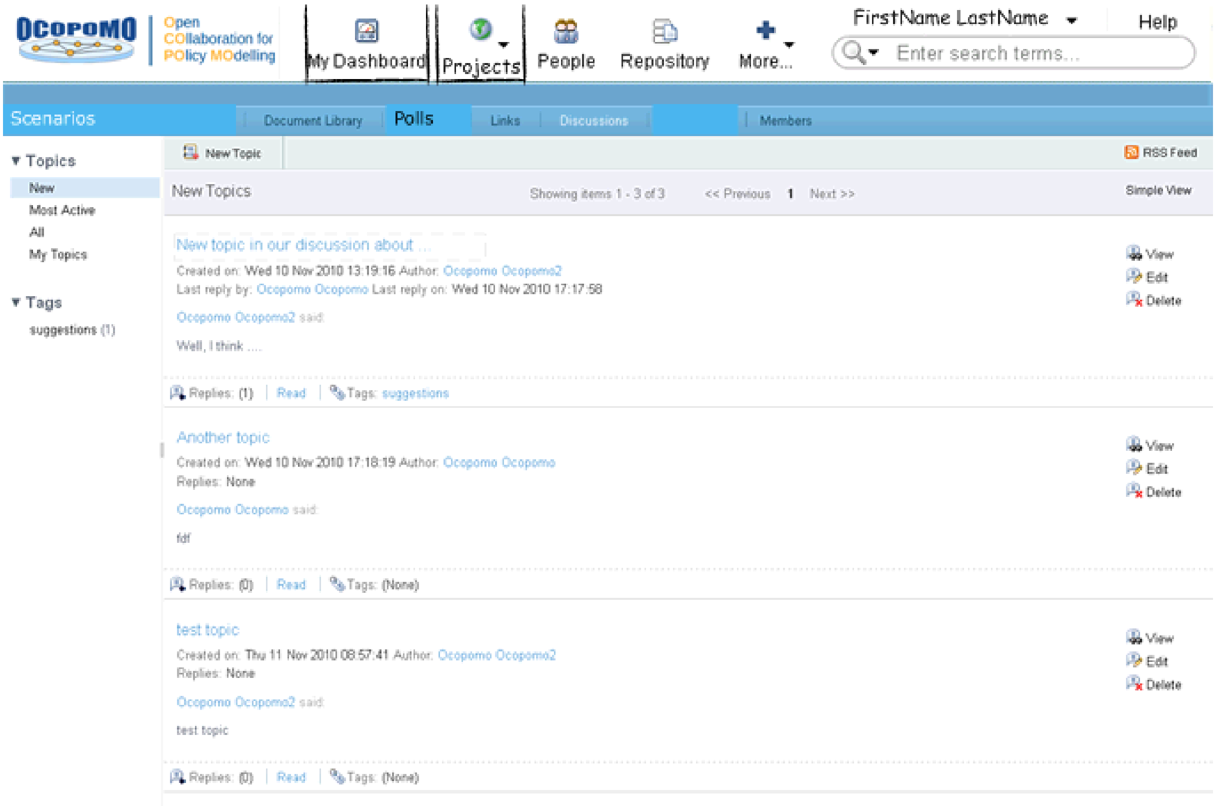
Window opens by mouse click on „contact author(s)"-button

Figure 75 Mock-up for contacting authors of the scenario



With it, people will be invited to the project and receive special access rights for the scenario (only reading and discussing; reading and writing and discussing).

Figure 76 Mock-up for inviting people to join the scenario generation



New Topics

Showing items 1 - 3 of 3 << Previous 1 Next >>

New topic in our discussion about ...

Created on: Wed 10 Nov 2010 13:19:16 Author: Ocopomo Ocopomo2
Last reply by: Ocopomo Ocopomo Last reply on: Wed 10 Nov 2010 17:17:58

Ocopomo Ocopomo2 said:

Well, I think ....

Replies: (1) | Read | Tags: suggestions

Another topic

Created on: Wed 10 Nov 2010 17:18:19 Author: Ocopomo Ocopomo
Replies: None

Ocopomo Ocopomo said:

fdf

Replies: (0) | Read | Tags: (None)

test topic

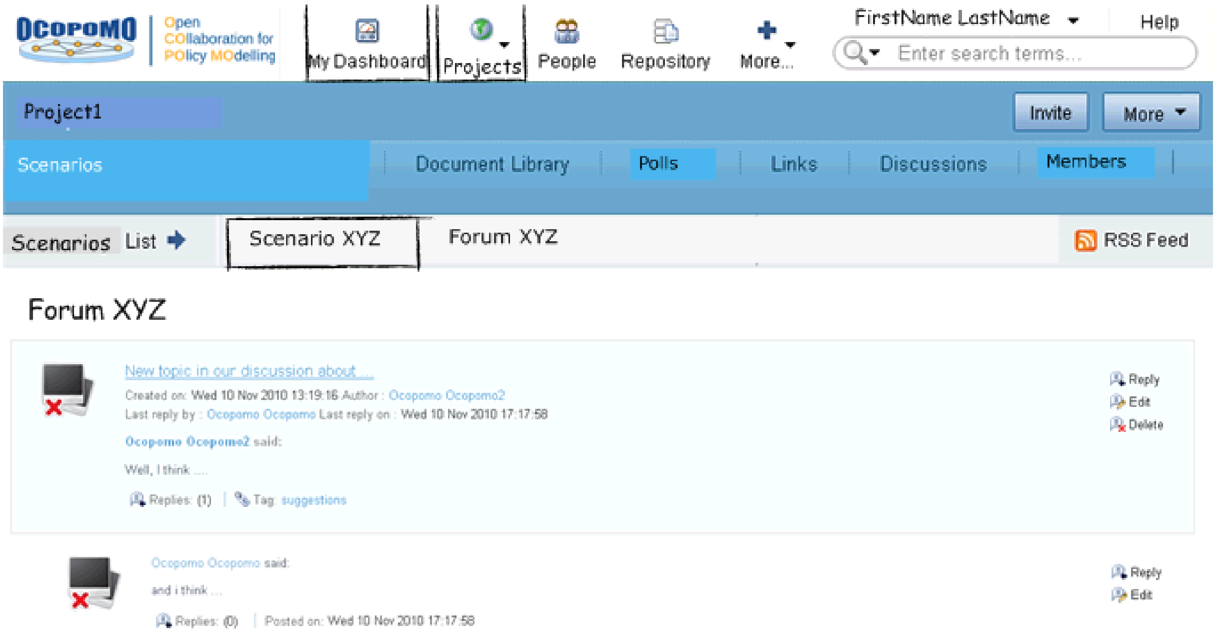
Created on: Thu 11 Nov 2010 08:57:41 Author: Ocopomo Ocopomo2
Replies: None

Ocopomo Ocopomo2 said:

test topic

Replies: (0) | Read | Tags: (None)

Figure 77 Mock-up for getting an overview of all scenario-related discussions structured by topic



Forum XYZ

New topic in our discussion about ...

Created on: Wed 10 Nov 2010 13:19:16 Author : Ocopomo Ocopomo2
Last reply by : Ocopomo Ocopomo Last reply on : Wed 10 Nov 2010 17:17:58

Ocopomo Ocopomo2 said:

Well, I think ....

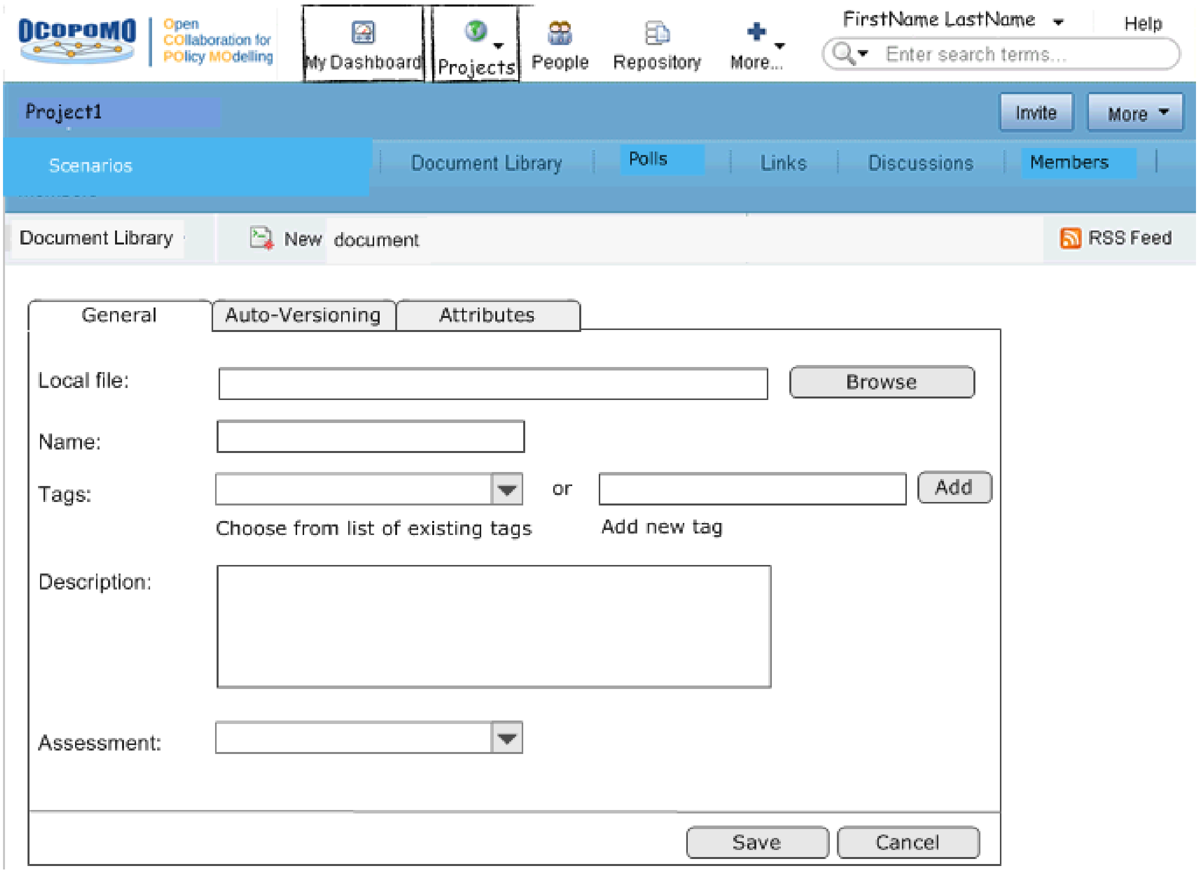
Replies: (1) | Tag: suggestions

Ocopomo Ocopomo said:

and i think ...

Replies: (0) | Posted on: Wed 10 Nov 2010 17:17:58

Figure 78 Mock-up for viewing and contributing to a specific scenario-related discussion



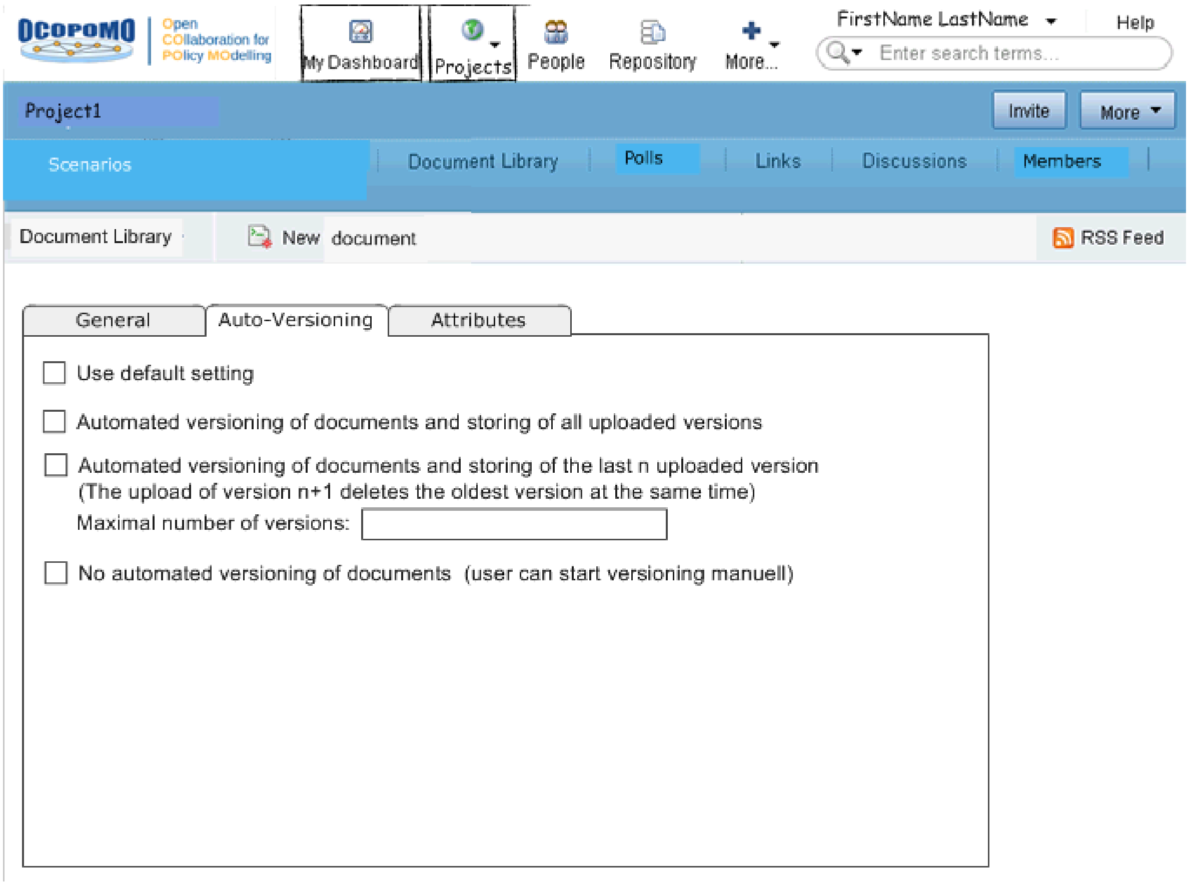
Local file:

Name:

Tags: Choose from list of existing tags or Add new tag

Description:

Assessment:

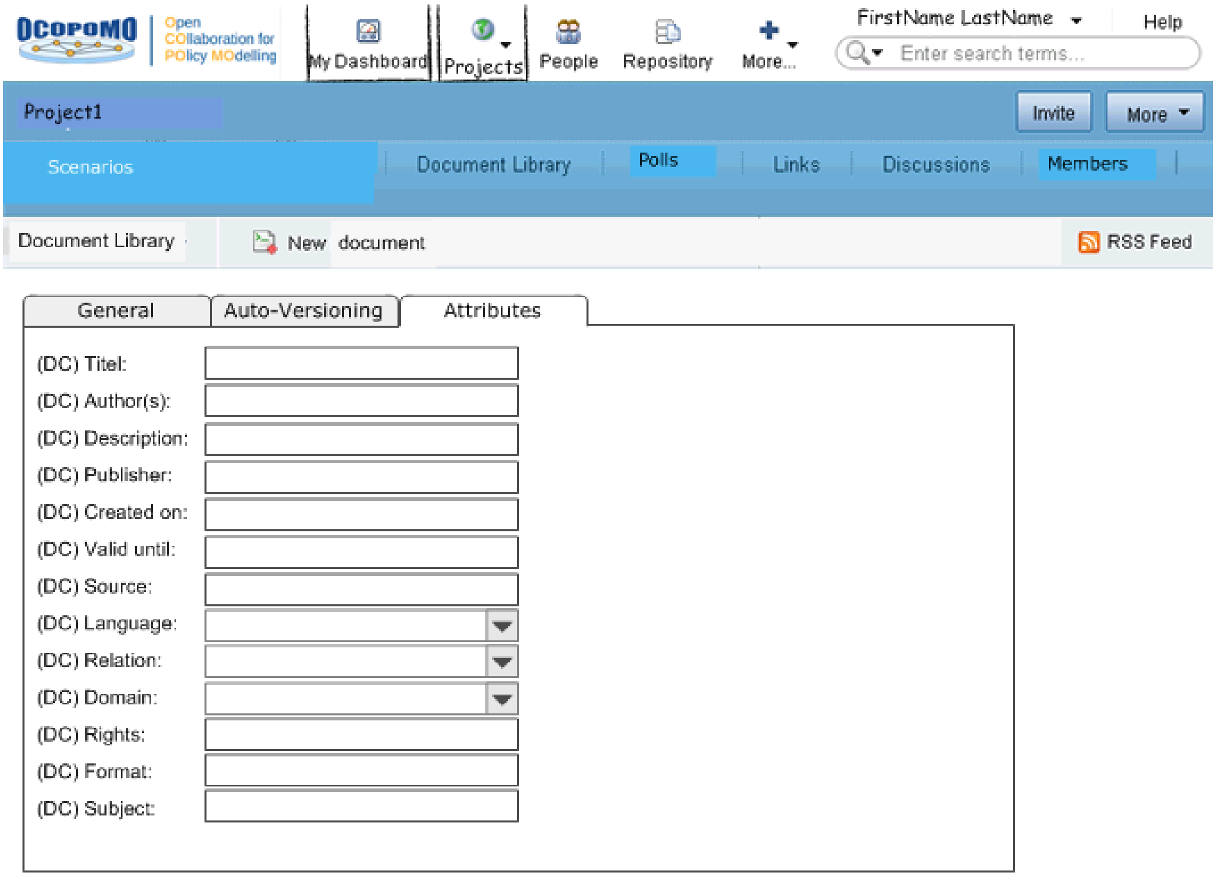


☐ Use default setting

☐ Automated versioning of documents and storing of all uploaded versions

☐ Automated versioning of documents and storing of the last n uploaded version
(The upload of version n+1 deletes the oldest version at the same time)
Maximal number of versions:

☐ No automated versioning of documents (user can start versioning manuell)



(DC) Titel:

(DC) Author(s):

(DC) Description:

(DC) Publisher:

(DC) Created on:

(DC) Valid until:

(DC) Source:

(DC) Language:

(DC) Relation:

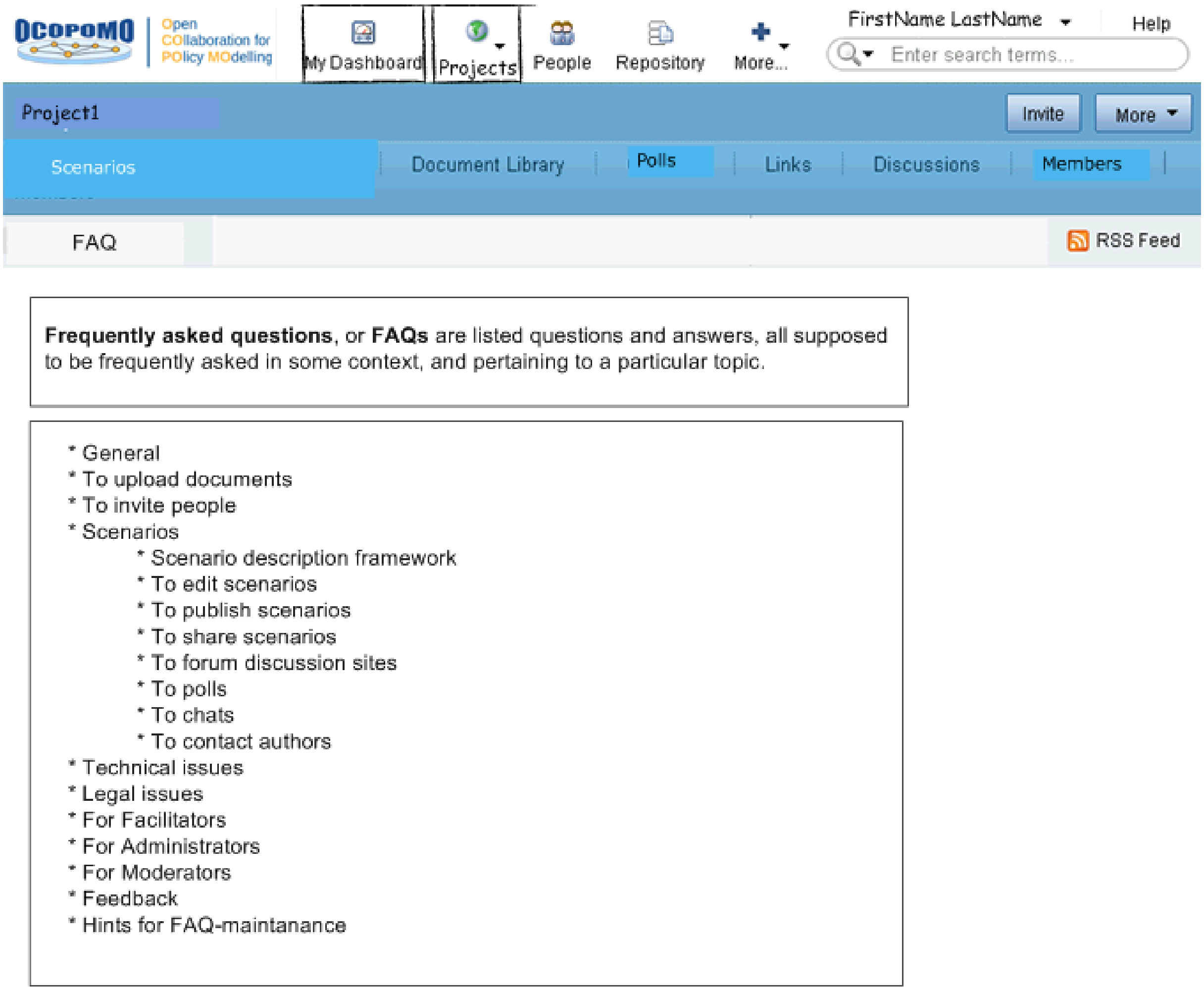
(DC) Domain:

(DC) Rights:

(DC) Format:

(DC) Subject:

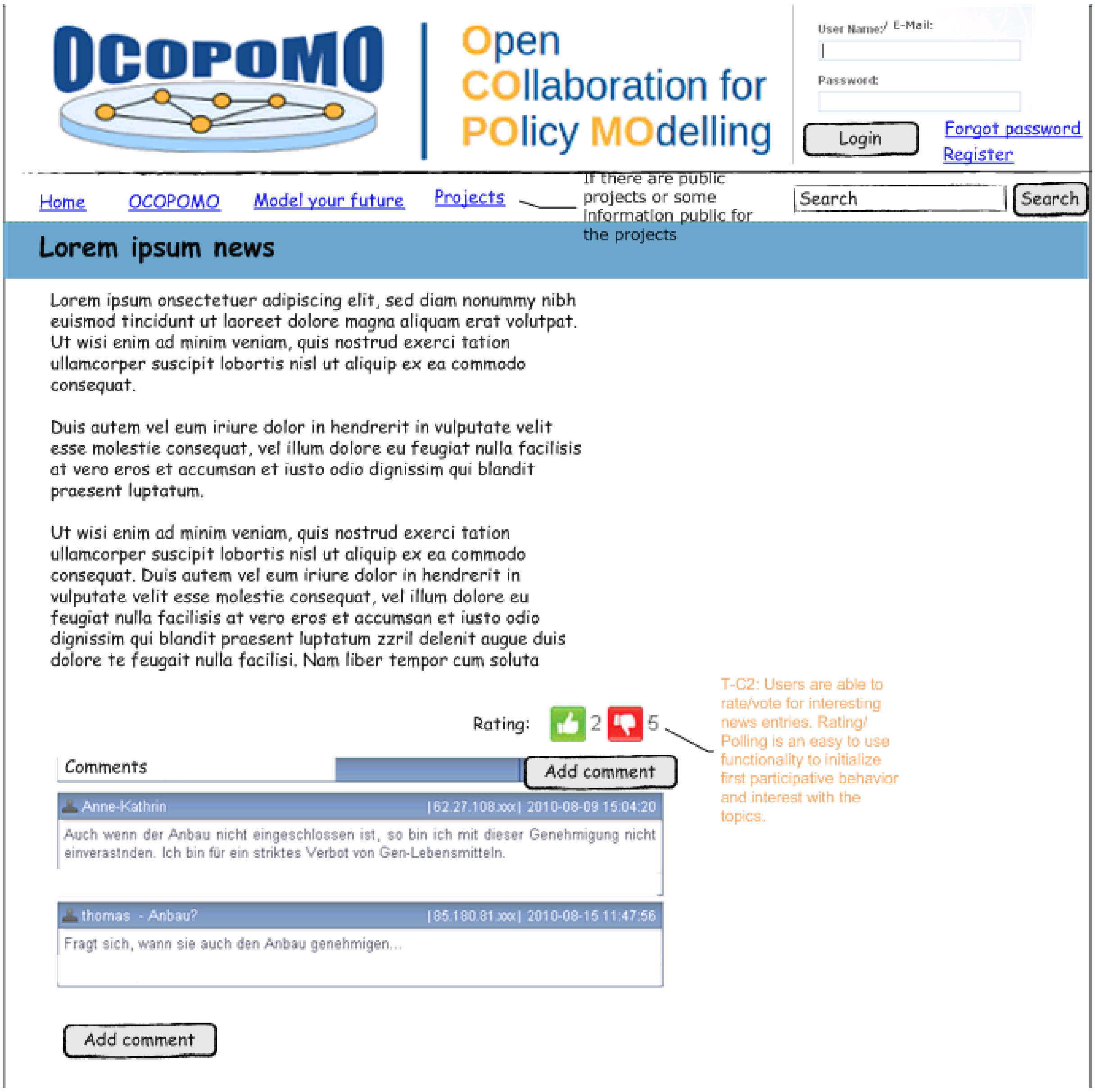
Figure 79 Upload documents, inserting data about document and setting conditions



Frequently asked questions, or **FAQs** are listed questions and answers, all supposed to be frequently asked in some context, and pertaining to a particular topic.

* General
* To upload documents
* To invite people
* Scenarios
  * Scenario description framework
  * To edit scenarios
  * To publish scenarios
  * To share scenarios
  * To forum discussion sites
  * To polls
  * To chats
  * To contact authors
* Technical issues
* Legal issues
* For Facilitators
* For Administrators
* For Moderators
* Feedback
* Hints for FAQ-maintanance

Figure 80 Mock-up for Frequently Asked Questions (FAQ)



Lorem ipsum news

Lorem ipsum onsectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.

Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum.

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Nam liber tempor cum soluta

Rating: 2 5

T-C2: Users are able to rate/vote for interesting news entries. Rating/ Polling is an easy to use functionality to initialize first participative behavior and interest with the topics.

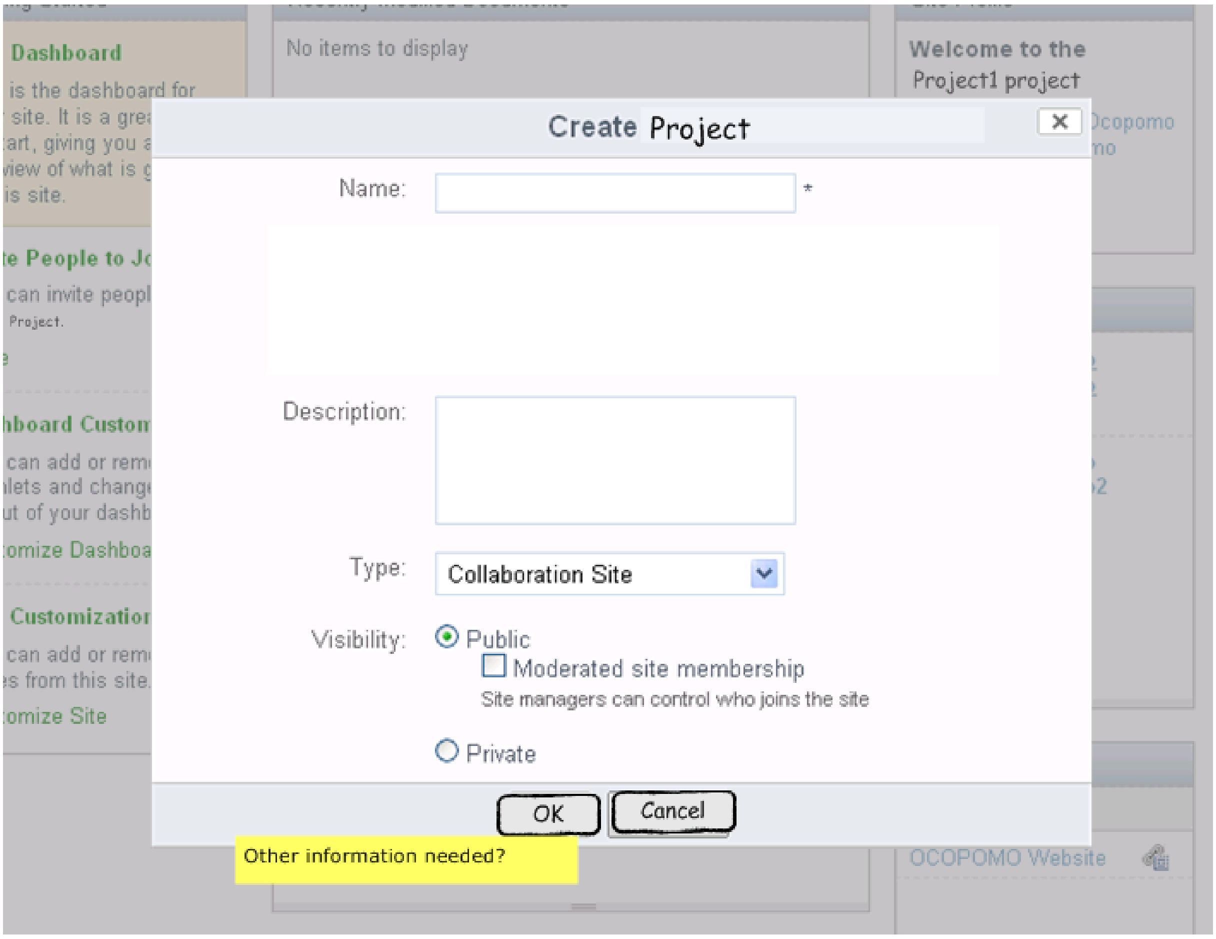
Anne-Kathrin | 62.27.108.xxx | 2010-08-09 15:04:20

Auch wenn der Anbau nicht eingeschlossen ist, so bin ich mit dieser Genehmigung nicht einverastnden. Ich bin für ein striktes Verbot von Gen-Lebensmitteln.

thomas - Anbau? | 85.180.81.xxx | 2010-08-15 11:47:56

Fragt sich, wann sie auch den Anbau genehmigen...

Figure 81 Mock-up for news entry



Create Project

Name: *

Description:

Type: Collaboration Site

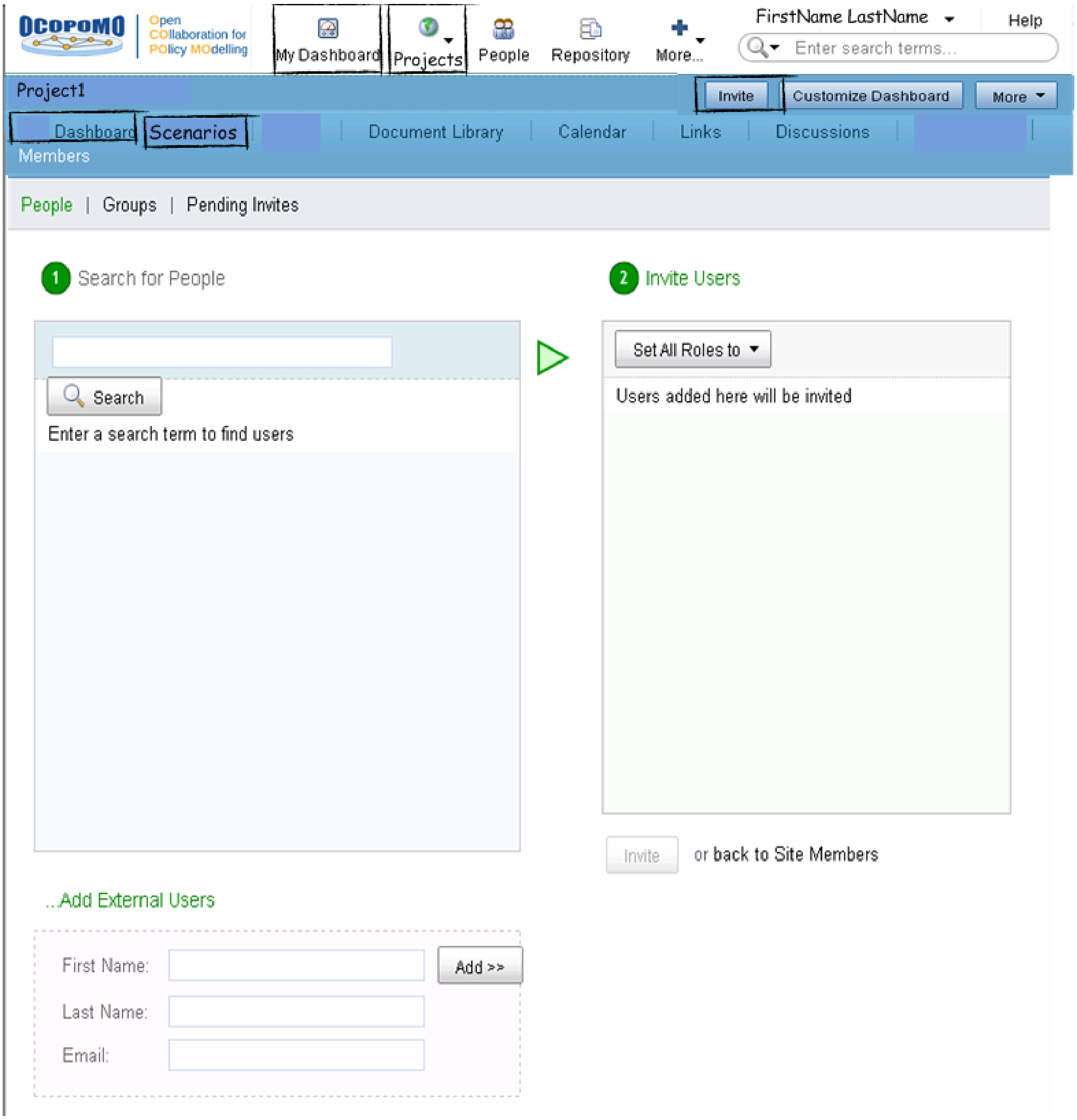
Visibility: ● Public

□ Moderated site membership

Site managers can control who joins the site

○ Private

Figure 82 Mock-up for creating a project



Enter a search term to find users

Users added here will be invited

Invite or back to Site Members

First Name:

Last Name:

Email:

Figure 83 Mock-up for inviting to a project

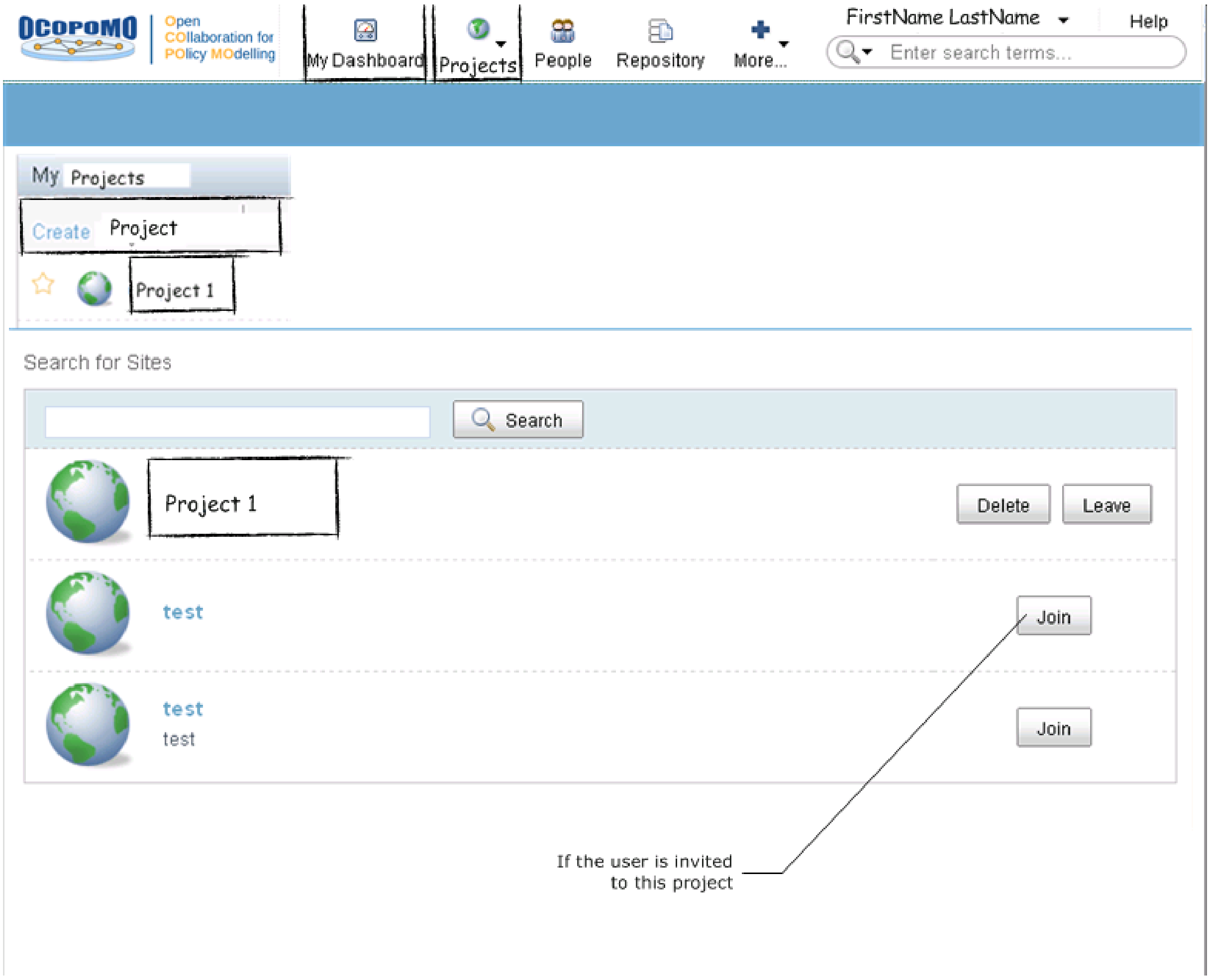


Figure 84 Mock-up for projects overview

APPENDIX D: SPECIFICATION OF DATA OBJECTS BASED ON USER REQUIREMENTS

User requirements, provided in [Bicking et al., 2010], were analysed to identify proper data objects and information resources for the OCOPOMO platform. The results of this analysis are presented in the table below. The table columns, from left to right, contain:

1. Identifier and name of the main requirement (i.e. which initially invoked a definition of specific information resource or data type) together with its type and priority. Optionally, a list of other relevant requirements is included.
2. Identification of the information resources proposed to store and maintain the data (information) for the required functionality.
3. Identification of particular data objects within the information resources. The name of a data element is marked in boldface.

The resulting structure of data objects, their relationships and distributions within the information resources are described and discussed in more details in section 6.2.2.

Requirement ID & name, type & priority:	**Identified:**	
	Information resources	**Data objects**
T-1 Discussion forums Functional, Must-have (includes T-1-1, T-1-2, T-1-3, T-1-4, T-1-5, T-12, T-14)	▪ **e-Participation information resources** (ePartIR) – **Discussions**. Space for persistent storage of discussions and their elements. ▪ **User management, security**. Authentication and authorization data for users. ▪ **CMS – Context**. Linking, capturing the context of published information. Relations of discussions to scenarios, policy models, documents, chat, etc.	▪ **discussion forum**: discussion, thread, topic of interest (a reference to scenario, policy model, document, or other resource type), conditions of use (moderated / non-moderated), attached “Rules for engagement”, context (related information resources) ▪ **discussion contribution**: content, properties (date/time, contributor), rating ▪ **user**: discussion moderator, contributor
T-4 Chat Functional, Must-have	▪ **ePartIR – Chat**. Space for persistent storage of chats, on-line discussions. ▪ **User management, security**. Authentication and authorization data for users. ▪ **CMS – Context**. Relations of a chat to scenarios, policy models, documents, discussions, etc.	▪ **chat**: content, properties, status, history, context (related information resources) ▪ **user**: chat user, moderator
T-5 CMS functionality	▪ **CMS**. Space for persistent	▪ **document**: content, format,

Functional, Must-have	storage, access, and publishing of documents, including their versions. ▪ **User management, security**. Authentication and authorization data for users. ▪ **CMS – Context**. Relations of published documents to scenarios, policy models, other documents, discussions, chat, etc.	properties, versions, context (related information resources) ▪ **document flow**: creation, editing, templates, access rights ▪ **user**: doc creator, editor
T-7 Opinion polling tool Functional, Must-have (includes T-8, T-9, T-10, T-11; integrated requirement I-10)	▪ **ePartIR – Opinion polling** (OP). Space for persistent storage, access, and publishing of opinion polls, including their versions. ▪ **User management, security**. Authentication and authorization data for users. ▪ **CMS – Context**. Relations of an opinion polling to scenarios, policy models, documents, discussions, chat, etc.	▪ **opinion poll**: settings (time interval, participants, percentage of the filled in forms), status, polling subject (a reference to scenario, policy model, document, or other resource type), versions (possibility to modify the answers), history, results ▪ **OP question**: question type (multi-choice, text-based, etc.) ▪ **OP answer**: answer type, version, history ▪ **user**: authorized OP creator, OP participant
T-16 Agent-based simulation tool Functional, Must-have (includes T-17, T-18, T-19, T-20, T-21 (Should-have), T-22, T-23; non-functional requirements T-35, T-36, NFR03-PM, NFR04-PM, NFR05-PM, NFR07-PM, NFR08-PM, NFR09-PM, NFR10-PM, NFR11-PM, NFR12-PM, NFR13-PM, NFR14-PM; integrated requirements I-18, I-24, I-25, I-26, I-27, I-28)	▪ **Simulation Model** (SM). Space for persistent storage, access, and publishing of policy models and related simulations. ▪ **User management, security**. Authentication and authorization data for users. ▪ **CMS – Context**. Relations of a policy model to scenarios, documents, discussions, chat.	▪ **simulation model**: content, state, properties / parameters, agents, rules, versions ▪ **SM agent**: ID/name, properties ▪ **SM rule**: content, properties ▪ **simulation**: content, events, properties, context (related scenarios, documents, etc.), level of details, time scale, cycle No., related policy model ▪ **SM event**: content, type, properties ▪ **scenario**: ID/name, document, properties, versions, reference to a policy model ▪ **user**: authorized PM creator, PM editor, PM participant, policy stakeholder
T-24 News functionality Functional, Must-have	▪ **ePartIR – News**. Space for persistent storage, access, and	▪ **news**: content, format, properties, context (related

(includes T-C2)	publishing of news. ▪ **CMS – Context**. Relations of news to scenarios, policy models, documents, discussions, chat, opinion polling, etc.	information resources), rating ▪ **user**: news creator / editor / publisher
T-25 Commenting functionality Functional, Must-have	▪ **ePartIR – Comments**. Space for persistent storage, access, and publishing of comments. ▪ **CMS – Context**. Relations of comments to scenarios, policy models, documents, discussions, chat, etc.	▪ **comment**: content, format, reference to commented resource, properties, context (related information resources) ▪ **user**: authorized creator / editor of comments
T-28 Calendar Functional, Should-have	▪ **ePartIR – Calendar**. Space for persistent storage, access, and automatic publishing of events (based on date/time settings). ▪ **CMS – Context**. Relations of calendar events to scenarios, policy models, documents, discussions, chat, etc.	▪ **calendar**: events, properties ▪ **event**: date/time settings of validity, properties, context (related information resources) ▪ **user**: authorized creator / editor of events
T-29 Newsletter Functional, Must-have	▪ **ePartIR – Newsletter**. Space for persistent storage, access, and publishing of newsletter documents. ▪ **CMS – Context**. Relations of a newsletter to scenarios, policy models, documents, discussions, chat, etc.	▪ **newsletter**: content, format, properties, context (related information resources), means of delivery list of recipients / subscribers ▪ **e-mail notification**: properties (sender, addressee, subject,...) ▪ **user**: newsletter creator / editor / publisher, newsletter recipients / subscribers
T-30 RSS Functional, Must-have	▪ **ePartIR – RSS**. Space for persistent storage, access, and publishing of RSS representations of information.	▪ **RSS feed**: XML content, references to related information resources
T-34 E-mail notifications Functional, Must-have	▪ **ePartIR – E-mail notifications**. Space for persistent storage and access of e-mail messages sent to participants.	▪ **e-mail notification**: properties (sender, addressee, subject,...), awareness frequency (daily/weekly/monthly) ▪ **user**: notification creator / editor, notification recipients / subscribers
T-38 Transcription tool Functional, Should-have	▪ **ePartIR – Transcription**. Space for persistent storage, access, and publishing of communication transcriptions (audio, video).	▪ **transcription**: properties, format, reference to transcribed information resource
T-39 Computer-assisted Qualitative Data	▪ **CMS – Textual Data Analysis** (TDA). Space for persistent	▪ **document**: content, format, properties, versions, context

Analysis Software Tool Functional, Must-have (includes T-40 (Should-have), T-41 (Nice-to-have), T-42 (Should-have) , T-43 (Nice-to-have))	storage and access of text phrases and their annotations (using meta-data tags). ▪ **CMS – Context**. Relations / links of text phrases (passages) to information resources (scenarios, policy models, other documents, discussions, chat, etc.).	(related information resources) ▪ **text phrase**: ID, position in the text (start, length), properties ▪ **meta-data tag**: name, properties, related tags ▪ **tag vocabulary**: list (hierarchy, structure) of tags ▪ **user**: doc editor / annotator
FR01_PM Policy model transformation Functional, Must-have (includes FR02_PM - FR27_PM; integrated requirements I-2, I-3, I-4, I-5, I-7, I-11, I-12, I-13, I-14, I-15, I-17, I-23)	▪ **CMS – Workspace**. Integrated data structure of policy models and e-Participation tools for scenario generation. ▪ **Narrative scenario**. Space for persistent storage, access, and publishing of text-based narrative scenarios, related to a policy model. ▪ **SM**. Space for persistent storage, access, and publishing of policy models and related simulations. ▪ **CMS – Context**. Relations of narrative scenarios and PMs to documents, discussions, chat, etc.	▪ **workspace**: ID, scenario (together with related policy models), ePartIR resources (discussions, opinion polls, etc.), users (stakeholders), properties ▪ **scenario**: ID/name, document, properties, versions, reference to a policy model ▪ **context**: parent scenario, references to related resources (policy models, documents, discussions, etc.) ▪ **simulation model**: content, state, properties / parameters, versions, environment (i.e. environmental aspects of the descriptive scenario), assumptions (minimal set of assumptions the model should carry) ▪ **SM event**: content, type, properties ▪ **SM rule**: : content, properties, clauses, related / dependent rules ▪ **SM agent**: ID/name, properties, reference to an user ▪ **user**: policy stakeholder, facilitator
TP-1 Rules in policy models Functional, Must-have (includes TP-2; integrated requirements I-39; I-40)	▪ **SM – Rules**. Space for persistent storage and access of rules defined for a policy model. ▪ **CMS – Textual Data Analysis** (TDA). Space for persistent storage and access of text phrases and their annotations (using meta-data tags).	▪ **SM rule**: content, properties, clauses, related / dependent rules ▪ **SM rule clause**: content, properties, related clauses ▪ **language translation**: clause, text phrase ▪ **text phrase**: ID, position in

		the text (start, length), properties
TP-3 Outputs of policy modelling Functional, Must-have (includes TP-5)	▪ **SM – Output**. Space for textual outputs generated by a simulation. The output is a type of document, which belongs to the TDA. It can also be used as an initial narrative scenario.	▪ **SM output**: content, format, properties, versions, reference to parent policy model
T-37 Authorization and authentication Non-functional, Must-have (includes integrated requirements I-F-I1, I-F-I2, I-F-I3, I-F-I4, I-F-I5, I-F-I6)	▪ **User management, security**. Authentication and authorization data for users.	▪ **user**: ID, properties, profile, roles, individual access rights, credentials ▪ **user profile**: properties / parameters, preferences ▪ **user role**: properties, role-based access rights, credentials, references to tools / modules
NFR01_PM Scenario description Non-functional, Must-have (includes integrated requirement I-22)	▪ **Narrative scenario**. Space for persistent storage, access, and publishing of text-based narrative scenarios, related to a policy model. ▪ **Narrative scenario – Consistent Conceptual Description** (CCD): structure of concepts (tags) and annotations describing a scenario. ▪ **CMS – TDA**. Space for persistent storage and access of text phrases and their annotations (using meta-data tags).	▪ **scenario**: ID/name, document, properties, versions, reference to a policy model ▪ **scenario CCD**: a structure of tags (concepts) and text phrases, extracted from the textual content of the scenario ▪ **text phrase**: ID, position in the text (start, length), properties ▪ **meta-data tag**: name, properties, related tags ▪ **tag vocabulary**: list (hierarchy, structure) of tags
NFR02_PM Language translation Non-functional, Should-have (includes NFR06-PM; integrated requirement I-30)	▪ **Narrative scenario – CCD language translation**: structures for mapping natural language words or phrases to the concepts (tags). ▪ **CMS – TDA**. Space for persistent storage and access of text phrases and their annotations (using meta-data tags).	▪ **scenario CCD**: a structure of tags (concepts) and text phrases, extracted from the textual content of the scenario ▪ **CCD mapping**: word / phrase, meta-data tag (concept) ▪ **meta-data tag**: name, properties, related tags ▪ **tag vocabulary**: list (hierarchy, structure) of tags
I-1 ICT toolbox as a portal-based web application Integrated, Must-have	▪ **Centralized data repository**: integrated data storage with a unified access, suitable for web applications.	▪ **data connector**: session, connection, user, data ID, data type, properties

I-2 Transformation table Integrated, Must-have	▪ **CMS – Context**. Space for persistent storage and access of relations (association links) between two information resources (e.g., between scenarios and policy models, between scenarios and documents, etc.).	▪ **context**: session, connection, user, data ID, data type, properties
I-5 Search Integrated, Must-have (includes I-6)	▪ **CMS – Search**. Space for persistent storage of indexed data (text-based documents, scenarios, meta-data, discussions, etc.); should allow quick access for search and data retrieval.	▪ **search index**: indexed data, data type, last indexed date, properties
I-19 Logs Integrated, Must-have (includes I-20, I-29)	▪ **CMS – Log**. Space for persistent storage of system logs, generated by various resources (scenario generation, transformation, user actions, etc.); should allow filtering and searching the stored data.	▪ **log index**: data ID, data type, actor / user, date / time, properties
I-32 Workflow support Integrated, Must-have	▪ **CMS – Workflow**. Space for persistent storage of workflow sequences, tasks and actions performed on various resources.	▪ **workflow**: ID, properties, structure of tasks ▪ **task**: ID, reference to parent workflow, inputs, outputs, preconditions, effects, used resources (documents, etc.)

Table 66 Data analysis of user requirements